WWU
MÜNSTER

Gamification for Business Process Modeling

Nicolas Pflanzl

Gamification for Business Process Modeling

Inauguraldissertation zur Erlangung des akademischen Grades eines Doktors der Wirtschaftswissenschaften durch die Wirtschaftswissenschaftliche Fakultät der Westfälischen Wilhelms-Universität Münster

Vorgelegt von

Nicolas Pflanzl, MSc
aus Wien

Mai 2017

Dekanin	Prof. Dr. Theresia Theurl
Erster Gutachter	Prof. Dr. Gottfried Vossen
Zweiter Gutachter	Prof. Dr. Patrick Delfmann
Dritter Gutachter	Prof. Dr. Jens Leker
Mündliche Prüfung	30.06.2017

Nicolas Pflanzl

Gamification for Business Process Modeling

Wissenschaftliche Schriften der WWU Münster

Reihe IV

Band 13

Nicolas Pflanzl

Gamification for Business Process Modeling

Wissenschaftliche Schriften der WWU Münster
herausgegeben von der Universitäts- und Landesbibliothek Münster
http://www.ulb.uni-muenster.de

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Dieses Buch steht gleichzeitig in einer elektronischen Version über den Publikations- und Archivierungsserver der WWU Münster zur Verfügung.
http://www.ulb.uni-muenster.de/wissenschaftliche-schriften

Nicolas Pflanzl
„Gamification for Business Process Modeling“
Wissenschaftliche Schriften der WWU Münster, Reihe IV, Band 13
Verlag readbox publishing GmbH – readbox unipress, Münster
http://unipress.readbox.net

Zugl.: Diss. Universität Münster, 2017

ISBN 978-3-8405-0176-0 (Druckausgabe)
URN urn:nbn:de:hbz:6-29169441136 (elektronische Version)

direkt zur Online-Version:

Satz: Nicolas Pflanzl
Titelbild: Nicolas Pflanzl
Umschlag: ULB Münster

Preface

Business process management (or BPM for short) is an "old" area in the field of information systems, as it goes back to the work of Michael Hammer and James Champy on Reengineering the Corporation that appeared almost 25 years ago. Since the early 1980s, the field has experienced both widespread adoption in industry and serious investigation in academia, which has led to a host of process models and corresponding modeling methodologies. Prominent models include event-driven process chains (EPCs), Petri nets, or the Business Process Model and Notation (BPMN) (to mention just a few), each of which has its community of "fans" or users, and often one is preferred over the other for various reasons. Petri nets, which emerged from the dissertation of German mathematician Carl Adam Petri in the 1960s, stand out for their notational clarity and their precise semantics, which even makes them amenable to mathematical foundations (e. g., in logic or in algebra).

In the past the practice of BPM has often seen a kind of "reengineering" in the sense that enterprises have a desire (or need) to adapt to modern technology or the digital age, and then recognize that they are not even fully aware of their business processes. In this situation, the resolution often is to call upon consultants or experts who analyze what is happening in the various departments of the enterprise and then try to combine their findings into a "big picture." Experience shows that the results of such an approach are often disappointing, since the process models produced do not adequately reflect how a department is working (or how workers perceive their tasks); moreover, the result is often neither providing a basis for process optimization, in most cases another BPM goal. In other words, in many cases there is a gap between what process modelers construct and how reality is perceived by those who actually execute the processes under consideration.

Various attempts have been made over the years to close this gap. One of them is BPM socialization, or the idea of bringing process modelers and process users together in a modeling project and to do so in an early stage of the project. This approach has various challenges for users, e. g., unfamiliarity with the tools and techniques employed, fear of making mistakes, fear of outing oneself as incompetent, or simply being unable or unwilling to learn BPM, etc. A way to overcome these obstacles, at least in many cases, is the provisioning of incentives, and this is where gamification enters the picture. In a nutshell, gamification is the idea of (re-)using gaming concepts in a non-gaming context. Nicolas Pflanzl's dissertation is the first comprehensive (and successful) attempt to study how gamification can be used to support BPM and to actually integrate gamification into a BPM tool.

His work is structured into three parts, where he starts with an introduction to the core elements of BPM and to gamification, which first puts this topic into context and then considers game design elements as well as theoretical foundations. In the main part of the thesis, Pflanzl opens with a problem specification that clearly identifies the challenges of social BPM as well as the potentials that gamification might exhibit in this context. After a brief description of the context in which his concrete developments will be settled, he embarks on the design of his specific concept. His gamification design concept focuses on points, badges, and leaderboards.

Pflanzl has not only developed a collection of concepts to fuse BPM and gamification which may or may not be useful; he has additionally been able to realize this concept in the context of a BPM tool that is generally and freely available (i. e., not just a university prototype!). Indeed, Pflanzl proves that he not only understands economic concepts or psychological ones; he is also a true computer scientist. In particular, he is a profound programmer who is able to find his way into an existing (and complex) software system, identify the places where his contributions should be docking, and integrate it in a way that is minimally invasive to the original tool and its concepts. A tool that is new and that is based on the idea that the fusion of two previously distinct domains may be beneficial requires some form of proof-of-concept. To this end, Pflanzl has been able to evaluate his approach comprehensively. He has performed two different studies, a field study in the context of our "Introduction to Information Systems" course for first-year students, and a lab experiment with participants recruited from that course. For both studies, the important and fundamental question obviously is: Does gamification do any good in BPM? This question is answered in a way I am not foreclosing here; it suffices to say that Pflanzl discusses his findings, but also his limitations and threats to the validity of his results very frankly.

Nicolas Pflanzl's dissertation is the first comprehensive (and successful) attempt to study how gamification can be used to support BPM and to actually integrate gamification into a BPM tool. The dissertation is a beautiful piece of work which is well and clearly written throughout, clearly structured, and which keeps its central theme constantly, but also critically in focus. The thesis is also a nice fusion of information systems and computer science capabilities, and he also brings in the necessary and relevant statistics and statistical methods when it comes to taking a critical look at what he has been doing in terms of evaluating his accomplishments. With this, the thesis is a highly interesting read for people working (or planning to work) in these fields, and I hope it will gain a wide audience.

Münster, December 2017 Prof. Dr. Gottfried Vossen

Acknowledgments

First and foremost, I would like to thank my doctoral adviser Prof. Dr. Gottfried Vossen for his support and guidance throughout my research and the process of writing this thesis. I am especially thankful that he demonstrated his belief in my capabilities by encouraging me to conduct research at the intersection of gamification and BPM, a novel fusion of topics without significant prior work. Moreover, I am grateful for the numerous opportunities that he gave me to present my work at national and international conferences, and to visit Brazil twice to conduct collaborative research with Prof. Dr. Renata Araujo. I would also like to thank Prof. Dr. Patrick Delfmann for his willingness to act as a second reviewer of this thesis, and Prof. Dr. Jens Leker for acting as a third examiner during the defense of this thesis.

Furthermore, I want to express my sincerest gratitude to Dr. Frank Schönthaler and Dr. Thomas Karle, who allowed me to conduct my doctoral research as part of a real-world project. Furthermore, I would like to thank all employees of PROMATIS software GmbH and Horus software GmbH, in particular Kai Hussong for being my most vocal advocate and supporter, Anna Schmidt for designing the badge icons used in Chapter 6, Johannes Michler and Dr. Mana Taghdiri for providing me with amicable, patient, and professional support throughout the entire implementation, and Torsten Rieger for playing a significant part in the realization of an important component of the implementation.

During my time at the DBIS Group, I received invaluable support from my colleagues who contributed to my thesis in various ways. In no particular order, I want to thank Dr. Florian Stahl for setting a good example on how to complete a successful doctorate, David Fekete and Fabian Schomm for proofreading the entirety of my thesis and providing thoughtful feedback, Dr. Jens Lechtenbörger for scrutinizing my formal definition of business process graphs, Katharina Hoffmann for digitizing the results of the questionnaires used in the laboratory experiment, Lena Hertzel for reviewing parts of my bibliography, Denis Martins for giving me pointers about hypothesis testing, Ralf Farke for his competent and friendly technical support, and Claudia Werkmeister and her successor Melanie Wietkamp for their commendable assistance with all administrative tasks. Further contributions were made by the students Phil Godwin, Jannik Hoffjann, and Svetlana Dzgoeva. Lastly, I am grateful to Felix Nolte for taking over the reins of the "Horus gamification" project and wish him all the best for his doctorate.

When conducting academic research, valuable and sound results can only be obtained when working together with others. Therefore, I want to acknowledge my co-authors who have allowed me to broaden my horizon and develop various ideas that are reflected in my dissertation in one way or another. Specifically, I want to thank Dr. Armin Stein and Dr. Katrin Bergener for allowing me to test my ideas and demonstrate their effectiveness in the *Introduction to Information Systems* lecture, Prof. Dr. Renata Araujo and Tadeu Classe for their hospitality and our collaborative work on the use of serious games for citizen engagement, Dr. Mike Preuß and Vanessa Volz for involving me in their research on automated game balancing, Dr. Dominic Breuker, Dr. Hanns-Alexander Dietrich, Dr. Matthias Steinhorst, and Prof. Dr. Patrick Delfmann for our joint research on graph-based pattern matching in conceptual models, and Arthur Vetter, Prof. Dr. Andreas Oberweis, Dr. Frank Schönthaler, and Kai Hussong for the collaborative effort on a paper about the results of my doctoral project.

Finally, I am immensely grateful to my parents for supporting my decision to move to Münster for my studies in 2007, and to my wife's parents for welcoming me into my "second", German family with open arms. My heartfelt gratitude also extends to my friends—particularly Julia Constantin, Sebastian Luhn, and Katharina Hoffmann—who have helped me to maintain my mental and emotional stability throughout the process of writing this thesis. Last but certainly not least, I am eternally thankful to Lara Pflanzl, who started accompanying me on the journey to my doctorate as my girlfriend and became my wife along the way. She has always given me all the love, patience, and support that I could wish for and is one of the main reasons why I was able to bring this journey to a successful conclusion.

Münster, January 2018 Nicolas Pflanzl

Contents

List of Figures

List of Tables

1 Introduction

> "*Here we have at once a very important point: even in its simplest forms on the animal level, play is more than a mere physiological phenomenon or a psychological reflex. It goes beyond the confines of purely physical or purely biological activity. It is a significant function—that is to say, there is some sense to it. In play there is something "at play" which transcends the immediate needs of life and imparts meaning to the action. All play means something.*"
>
> [Hui49, p. 1]

1.1 Motivation

In his seminal book, *Homo Ludens*, Johan Huizinga claims that "play is older than culture, for culture, however inadequately defined, always presupposes human society, and animals have not waited for man to teach them their playing" [Hui49, p. 1]. Indeed, humans have played games for thousands of years, and one of the oldest known examples of a board game, *Senet*, dates as far back as 4600 years to ancient Egypt and bears a resemblance to modern-day *Backgammon* [ENST08]. Similarly, the histories of *Go* and *Chess*, two games still highly popular today, can be traced back to at least 2000 B. C. and 776 B. C. respectively. Digital games are a much newer phenomenon, and spurred by developments of Information Technology (IT) in the 20th century, their first precursors can be found as early as the 1950s. However, it is the game Spacewar! (Steve Russell, 1962) that can be seen as the prototype of modern computer games as they exist today. The 1970s saw digital games entering private homes, being played as *video* games on consoles connected to a television set. Finally, in the wake of the popular Pong (Atari, 1972), the video games industry emerged [ENST08].

For the last 30 years, the marketing of games has predominantly been focused on teenage boys, which has resulted in the deeply-rooted prejudice that games are (and should be) only played by kids [33, 34]. As these kids grew up, they did not stop playing games, and thus the stereotype of the "lone manboy" playing games alone in his basement emerged. However, an examination of the *demographic statistics* of gamers quickly reveals that this conflicts with reality. According to a study about gamers and the gaming industry in the United States (US) carried out by the Entertainment Software Association (ESA) in 2016 [Ent16], people playing games regularly (more than three hours a week) can be found in about 63% of all households. Furthermore, the average gamer nowadays is 35 years old, thereby disproving the assertion that video games are only the domain of kids and adolescents. The same holds for the prejudice that only men play games, with females making up about 41% of the gamer population. Similar figures can also be found for other countries, such as Germany [2, Bun16]. In addition, the games industry has also become a significant *economic* force, with gamers spending a substantial portion of their financial resources on games and game-related merchandise. For instance, estimates on the amount of money spent on gaming products in the United States and Germany in 2015 amount to 23.5 billion USD and 1.99 billion EUR respectively [Ent16, 2]. On a global scale, the games market was valued at approximately 99.6 billion USD in 2016, thus making it one of the largest industries in general [nG16].

Today, gamers voluntarily invest a considerable amount of time in their hobby. For instance, McGonigal estimates that the total amount of time that gamers across the globe played the online game World of Warcraft (Blizzard Entertainment, 2004) from 2004 up until 2011 could amount to as much as 5.9 million years [21]. Overall, the average gamer plays about 6.3 hours of video games a week [36], with a considerable portion—the so-called *hardcore gamers*—claiming an average weekly playtime of about 22 hours [37]. This leads to the obvious question whether this time is wasted, or if playing games can actually yield positive consequences. Early research on the impacts of computer and video games has often focused on its negative effects, most notably violence and aggression. Whether a correlation between the two actually exists is the subject of an ongoing, highly controversial debate, with some studies claiming a link between exposure to violent games and aggressive behavior, cognition, affect, physical arousal, and decreased empathy and prosocial behavior in children and adolescents regardless of cultural background and sex [AB01, And04, ASI+10], whereas other studies have found no such connection, asserting that previous results can partly be attributed to publication bias and other methodological shortcomings [Fer07a, Fer07b, FK10].

In recent years, attention has increasingly shifted towards the positive impacts of games. For instance, research has shown that games can (in some cases) help with knowledge acquisition, developing perceptual, cognitive, and motor skills, changing the behavior of players, and influencing motivation and affect [CBM+12]. This shift in the perception of games is not only reflected in academia, but also mainstream media. For instance, a recent article published in *The Guardian* provides seven reasons for why "grown ups" should play more digital games [34]: first, they are a very economical hobby and provide vast amounts of entertainment in relationship to their cost. Second, they serve as a preparation for the future of broadcast media, which are increasingly influenced by games. Third, they have become highly social and can be used to strengthen existing relationships when played together. Fourth, the democratization of development tools has made game creation a means for self-expression. Fifth, they are testbeds for many important technological developments, including virtual/augmented reality and artificial intelligence. Sixth, they allow connecting with kids, who have grown up as digital natives in a world permeated with games. And last but not least, they are fun.

Historically, play was considered a voluntary activity that is conducted in a safe environment clearly separated from reality—the so-called "magic circle" [SZ03]—and that has no consequences in the real world [Hui49, Cai61]. Thus, play was seen as inherently unproductive, i. e., unable to create any new goods, wealth, or other elements [Cai61]. However, such a clear separation was not always upheld in reality. For instance, the Kriegsspiel (Georg von Reisswitz, 1812) served the purpose of training Prussian army personnel in tactical battle maneuvers by letting them play a game that can be considered an ancestor of modern tabletop Role-playing Games (RPGs) [19]. Today, academics and practitioners are actively investigating whether the powers of digital games can be harnessed for purposes other than mere entertainment and with tangible consequences outside the magic circle. Some researchers even go as far as to claim that reality as it is today is inherently flawed and that games can serve as a guidebook for solving its problems [McG11]. This has led to the emergence of many different research directions, such as *serious games* [Abt70], *games with a purpose* [vA06], *game-based learning* [SZ03], and *pervasive games* [MSW09].

One of the newest manifestations of this type of research is *gamification*, the "use of game design elements in non-game contexts" [DDKN11, p. 9]. Discussion about gamification started as an industry-driven trend around 2008 and was picked up in academia briefly thereafter, with wide-spread publication starting about 2010 and gaining traction ever since [DDKN11, HKS14]. Today, it has surpassed the status of being simply the "newest buzzword" and is strongly represented at academic conferences in the areas of Computer Science, Human-Computer Interaction and Information Systems [SA15]. In contrast to the related research areas mentioned above, gamification is not concerned with the creation or utilization of full games for serious purposes, but with transferring individual elements presumed to be characteristic for games and game development to contexts such as health, work, and education [HKS14]. This includes, for instance, rewards such as points and badges, competition, tasks with clear goals, and story-driven narratives [TLB14]. Expected benefits of this approach include behavioral changes through increased motivation, engagement, and enjoyment of target users, and supporting learning processes [BL13]. The potential of gamification to achieve these goals has been confirmed in empirical studies, although results are generally mixed and the field has not yet reached a maturity where gamification can be applied with consistent, reliable results [HKS14, SF15b].

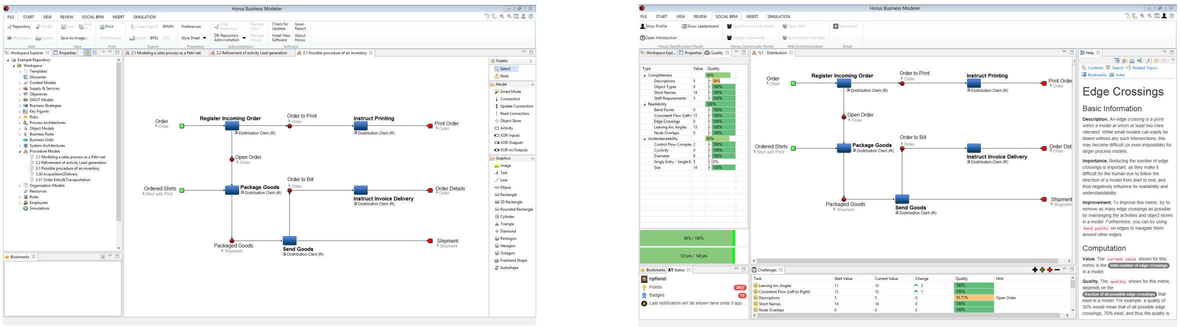
1.2 Research Question

In this doctoral thesis, the application of gamification to business process modeling is studied. Business process models are central to the design, implementation and enactment of business processes and thus serve as a valuable artifact for continuous process improvement. Business process modeling is the act of creating such models using visual, graph-based modeling languages and is often carried out as part of Business Process Management (BPM), a holistic approach for maintaining business performance by managing business processes [Ham10]. In academic literature and practice, modeling is often conceptualized as a task carried out by a small number of experts who elicit requirements from stakeholders through interviews and other tools, thereby confining the latter to a passive role [vBSR+14, RMH13, Ros06a]. However, a considerable number of publications question this understanding by describing the active involvement of process stakeholders as an important factor for successful BPM (see, e. g., [vBSR+14, AF08, BGR05]). By enabling process stakeholders to assume a more active role, modeling becomes a participative, collaborative effort undertaken by a larger and more heterogeneous set of participants [BGR05].

When coupled with the underlying principles of social software—collective intelligence, self-organization, egalitarianism, and social production—this leads to the notion of Social Business Process Management (Social BPM), a highly democratic manifestation of BPM in which practices emerge and are governed *bottom-up*, and all process stakeholders are empowered to participate [SN09, EGH+10]. Expected benefits of Social BPM are, e. g., an increased acceptance of the modeled—and eventually executed—business processes by their end-users, and better opportunities for leveraging potentials for process improvement and innovation [EGH+10]. However, a socialization of business process modeling comes with its own problems, such as motivating stakeholders to participate, enabling unexperienced novice modelers to contribute despite lacking the required skills, or ensuring the quality of the resulting process models [PV14]. Gamification may help with overcoming some of these challenges.

So far, only little work in the area of BPM can be found at the intersection of games and modeling. A notable exception is the use of 3D virtual worlds to support business process modeling in collaborative, distributed environments [BRW11]. Furthermore, BPM simulation games such as IBM INNOV8 [SJ09] exist. Lastly, some publications in the area of Social BPM discuss the use of "honor points" as a means for motivation [EGH+10]. However, a comprehensive analysis of the benefits that gamification may offer for business process modeling is missing, thus leading to the following exploratory research question: **how can gamification be used to support business process modeling?** From this question, more specific research goals are derived in Chapter 4.

The biggest opportunity for applying gamification to the context of BPM lies in adding game design elements to business process modeling software. Such digital tools are an essential part of business process modeling and support users in creating models with graphical editors and further advanced



(a) Without gamification. **(b)** With gamification.

Figure 1.1: Horus Business Modeler before and after the implementation of gamification.

functionality [Rec12b]. Gamification can be seen as such an advanced feature that allows addressing the challenges that arise in participative process modeling scenarios. For instance, the capability of game design elements to motivate and engage [HKS14] perfectly matches the requirement to increase the participation of process stakeholders. Consequently, the research question is examined through the conceptualization, implementation, and evaluation of a gamification module for the process modeling software *Horus Business Modeler (HBM)* to support participative modeling together in a heterogeneous community. Primary goals of this module are increasing the motivation of relevant stakeholders to actively and voluntarily participate in process modeling, ensuring that the resulting models have a high quality, and facilitating the acquisition of business process modeling skills while using the aforementioned tool. An outlook on the remainder of this dissertation is presented in Figure 1.1, which provides a side-by-side comparison between the Horus Business Modeler (HBM) before and after the integration of gamification.

1.3 Research Design

Design Science

In his seminal work on the "sciences of the artificial", Simon declares that "[everyone] designs who devises courses of action aimed at changing existing situations into preferred ones" [Sim96, p. 111]. Consequently, design as a science is concerned with improving the current state of reality through the conception and realization of valuable, innovative artifacts to achieve predefined goals [MS95, Sim96]. Therefore, it stands in stark contrast to the natural sciences which instead aim to understand reality as it exists to formulate theories about observable phenomena [MS95]. Within the context of the Information Systems (IS) discipline, carrying out design science research allows creating and evaluating IT artifacts through which identified organizational problems can be solved [HMPR04]. Following March and Smith, typical outcomes of design science research are the following artifacts [MS95]:

- **Constructs** are a conceptualization of a particular domain that can be used to describe problems within this domain and define potential solutions. Thus, constructs form the language and vocabulary that experts employ when thinking about pertinent tasks and sharing domain knowledge. Constructs can either be specified in a highly formalized manner, or in an informal fashion.
- **Models** can generally be described as representations of real or artificial originals that employ abstraction [Sta73]. Within the context of design science research, they form "[sets] of propositions or statements expressing relationships among constructs"[MS95, p. 256] that are "used to describe tasks, situations, or artifacts"[MS95, p. 253]. In contrast to the natural sciences, the main concern of models in design science is usefulness for particular users at a given time for a given purpose [Sta73] rather than objective truth.
- **Methods** build on constructs and models and can be defined as "[sets] of steps [...] used to perform a task"[MS95, p. 257] that can for instance be formalized as algorithms or guidelines. Methods are commonly used to solve problems by transforming models from one representation to another. They are of special importance for the natural sciences, which do not create methods but use those produced by design science research.
- **Instantiations** realize artifacts in their environments by operationalizing constructs, models, and methods. Thus, they "demonstrate the feasability and effectiveness of the models and methods they contain"[MS95,

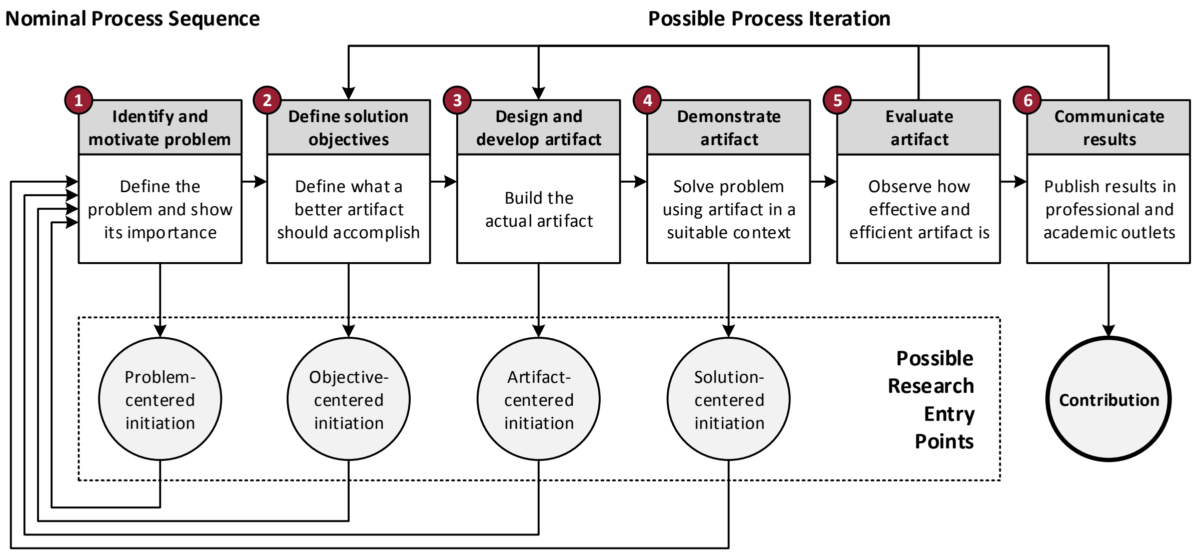


Figure 1.2: Design Science Research Methodology process. Source: based on [PTRC07, p. 54] as adapted by [Wei11, p. 50].

p. 258]. However, it may be that creating an instantiation must precede the formalization of the artifacts it is based on, as only its observation allows articulating constructs and other artifacts.

At its core, design science research consists of two basic activities: *building* and *evaluating* [MS95, HMPR04]. Whereas the former consists of constructing an artifact, thereby demonstrating that this is even possible, the latter refers to developing criteria against which the performance of the artifact can be measured, and performing the actual measurement [MS95]. Through this, researchers can answer the essential questions "does it work?" and "how well does it work?" It is common for this loop of building and evaluating to be iterated a number of times before the artifact reaches its final state. Furthermore, it is accompanied by two additional cycles: a relevance cycle ensuring that the artifact aims to solve a real problem and is capable of actually doing so, and a rigor cycle ensuring that the artifact builds on past knowledge and any research results are integrated back into the knowledge base [Hev07]. Based on these considerations and previous work such as seven guidelines for design science research in Information Systems proposed by Hevner et al. [HMPR04], a Design Science Research Methodology (DSRM) was proposed by

Peffers et al. [PTRC07]. Its accompanying nominal process is illustrated in Figure 1.2 and defines the following six activities:

① **Identify and motivate problem**. The first step of the research process consists of defining the research problem that should be addressed and justifying it by identifying the value of a potential solution. The latter is of special importance as it can motivate further researchers to also contribute to the problem and accept any research results that are disseminated. Part of the problem definition can also be a conceptual deconstruction of the research problem to derive requirements that should be addressed by the solution.

② **Define solution objectives**. Building on the problem specification, researchers must define the actual objectives the solution should achieve within the constraints of what is feasible and possible. Such objectives can either be quantitative (e. g., runtime improvements of a new algorithm) or qualitative (e. g., application scenarios enabled by a new instantiation). Performing this activity requires comprehensive knowledge of the current state of the research problem and solutions that currently exist (if any) as well as their performance.

③ **Design and develop artifact**. In this core step of the Design Science Research Methodology (DSRM), the actual artifact (i. e., a construct, a model, a method, or an instantiation) is created. Specifically, this involves determining the desired functionality of the artifact, its architecture, and its implementation. As March and Smith point out, these artifacts "then become the object of study"[MS95, p. 258]. As the space of potential solutions may be very large, it may be necessary to limit the search process to finding an acceptable alternative rather than the optimal one [Sim96]. Conceptually speaking, the output of this activity may be any "designed object in which a research contribution is embedded in the design"[PTRC07, p. 55].

④ **Demonstrate artifact**. After development has been finished, the effectiveness of the artifact must be demonstrated by successfully applying it to one or multiple problem instances. Depending on the nature of the artifact, this can be done using a variety of different methods such as case studies or experiments. During demonstration, the data that is required for the subsequent evaluation may be collected.

⑤ **Evaluate artifact**. Finally, the developed artifact must be observed and its performance measured to assess how well it is able to solve the specified problem. To that extent, the results collected during the demonstration of the artifact must be compared to the solution objectives defined in the second step. The exact type of evaluation that is appropriate depends on the nature of the artifact and the setting in which it was demonstrated and may take forms such as the quantitative analysis of objective performance metrics or of survey results. Generally speaking, this step may be comprised of any logical proof or appropriate means of empirical evidence. After evaluation, researchers may decide whether to iterate back to a previous activity if the performance of the artifact must be improved further.

⑥ **Communicate results**. Based on the results obtained from the evaluation, researchers should communicate "the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness"[PTRC07, p. 56] to academic as well as professional audiences at appropriate times. Typical outcomes of this step are research papers published at academic conferences or in academic journals.

This process is inherently iterative, meaning that there is no need to create the best-possible solution in the first attempt. Instead, researchers can go back from either of the last two activities to the definition of solution objectives or the design-and-development phase to incrementally improve their artifact. Furthermore, Peffers et al. acknowledge that a research endeavor may not always follow the nominal process sequence they propose, and thus allow for multiple research entry points [PTRC07]. In particular, besides a *problem-centered* approach starting with the definition of a problem, other possibilities for research initiation include an *objective-centered* approach motivated by needs expressed by practitioners or researchers, an *artifact-centered* approach starting with an already-existing artifact that has not yet been formally analyzed as a solution for the problem of interest, and a *solution-centered* approach where the effectiveness of an artifact has already been confirmed and research rigor is retroactively applied to its formation process.

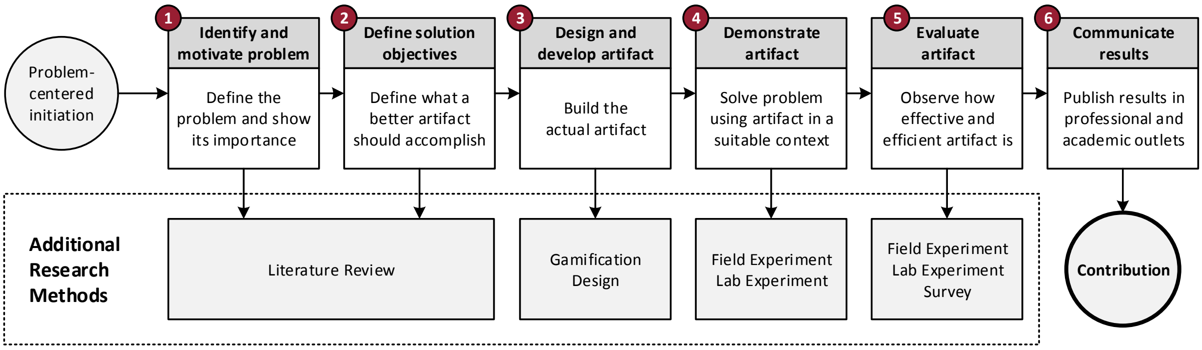


Figure 1.3: Applied Design Science Research Methodology.

This doctoral dissertation follows the DSRM proposed by Peffers et al. in its problem-centered manifestation [PTRC07], i. e., the research project was initiated with the identification and motivation of a relevant research problem as illustrated in Figure 1.3. Despite the iterative nature of the DSRM, the overall research process was only executed in a sequential fashion due to the constraints imposed by the project conditions. Nevertheless, the internal execution of certain activities—most notably within the design and development phase—was carried out in an iterative manner. In addition to the DSRM as an overarching research process, a number of complementary methods were used within particular activities. This is in line with the reasoning that different phases of a research project consist of varying tasks and problems whose individual requirements can best be met by combining multiple research methods [Min01]. Such considerations also play a role within design science. For instance, Hevner et al. distinguish 12 different methods that may be applied in the evaluation phase depending on the properties of the artifact and the defined solution objectives [HMPR04]. In the context of the research underlying this dissertation, the following supplemental methods were used:

- **Literature review**. A literature review allows creating the foundation on which any subsequent research activities are built and ensures that only relevant, open problems are addressed [WW02b]. Carrying out a review is a multi-stage process consisting of tasks such as defining the scope of the review, conceptualizing the topic of interest, searching for literature, analyzing and synthesizing the literature, and finally presenting the results to derive a research agenda [vBSN+09]. Arguably one of the core activities of a review, the search procedure can be supported through academic databases and performing backward and forward searches, i. e., examining articles referenced in a paper as well as articles citing a paper [WW02b]. In Information Systems (IS) literature, a number of proposals for carrying out *systematic*, *structured* literature reviews can be found [WW02b, LE06, vBSN^{+}09, OS10]. Despite the advantages of such a rigorous approach (e. g., reproducibility), the academic body of knowledge also contains a large number of high-impact reviews following an *unstructured*, *pragmatic* approach with undocumented review process, for instance the annotated bibliography on algorithms for drawing graphs by Di Battista et al. [DETT94], or the review of approaches for community detection in graphs by Fortunato [For10]. In the context of this research project, unstructured literature reviews were carried out in the first two phases of the design science process and published in [PV13b] and [PV14].

- **Experiments**. Experimental research allows studying cause and effect relationships by subdividing a set of respondents into two groups, of which one is exposed to a certain *treatment* (the experimental/treatment group), whereas the other (the control group) is not [Rec13]. For instance, in [RZSL06], the authors examined whether the willingness to pay of prospective buyers on the online auction platform eBay is higher for sellers with high reputation (the control group) than for sellers with low reputation (the experimental group). Depending on the setting, two types of experiments can further be distinguished: *controlled experiments* that are carried out in artificial scenarios that allow for a high degree of control, and *field experiments* that are conducted in real-world settings where conditions cannot be easily controlled [Rec13]. Overall, experiments are reportedly one of the most popular methods for conducting research in the IS discipline [OB91, CH04]. Both types of experiments also play an important role for this thesis and were used in the demonstration and evaluation phases of the design science process. A partial discussion of one field experiment was also published in [PBSV15b] and [PBSV15a], albeit with a focus outside the scope of this research project.

- **Survey**. A survey is “a means of gathering information about the characteristics, actions, perceptions, attitudes, or opinions of a large group of units of observations [...] referred to as a population" [Rec13, p. 76]. Generally speaking, surveys can be used to explore, describe, or explain the relationships between different variables [Neu14]. Furthermore, besides being used to advance scientific knowledge, surveys can also be found in practice, for instance as marketing surveys, opinion surveys, or political polls [PK93]. Conducting survey research is a multilevel process consisting of the development of a model of theoretical constructs and their expected relationships, the specification of how these constructs should be measured, the development and testing of the survey instrument, the administration of the actual survey, and lastly the analysis of the gathered data using statistical methods [Rec13]. The popularity of surveys in IS research exceeds that of all other research methods [OB91, CH04], and the same also holds for other disciplines such as the social sciences [Neu14]. Consequently, they were also employed in the evaluation phase of the design science process as a tool for assessing the performance of the developed instantiation.

- **Gamification design**. Due to the specific challenges of creating a gamified solution, the design phase of this research project utilized specific gamification design methods. While not technically a research method, the following subsection illustrates the strong correspondences between gamification and design science.

Gamification Design

Gamification sits at the intersection of a large variety of different disciplines and is for instance related to game design, behavioral psychology, human-computer interaction, Information Systems, and the concrete domain in which it is applied. Consequently, conducting gamification research is a demanding endeavor that requires an extensive set of skills and imposes specific requirements on the research process. For this reason, recent research has attempted to develop and (to some extent) validate more sophisticated gamification design methods (e. g., [Det13, Det15, MWHA17]) that go beyond previous proposals based on simple patterns and heuristic principles such as the popular (cf. [HKS14, NZT+14, MHK16]), yet controversial [32] Points, Badges, Leaderboards (PBL) approach or the "six Ds" of gamification proposed by Werbach and Hunter [WH12, p. 86]) that are discussed in Chapter 3. One such method was presented by Morschheuser et al. and is based on a rigorous review of literature and interviews with 25 gamification experts. By synthesizing their findings, the authors compiled the gamification process depicted in Figure 1.4 as a Unified Modeling Language (UML) activity diagram (cf. [Fow04]) that consists of the following steps:

① **Preparation**. The first step of a gamification project lies in clarifying which problem should be solved. To that extent, a *vision statement* outlining the core idea of the project should be created. This document can be used to communicate the main goals of the project to relevant stakeholders to ensure their support. Afterwards, a list of the *objectives* that should be achieved through the use of gamification should be compiled, ranked, and justified. Besides business goals, these objectives should also consider the needs and motivation problems of potential target users. Looking forward to the analysis step, objectives should be defined in a manner that allows for the derivation of measurable success metrics. The preparation phase finishes with the creation of a *project plan* in which requirements regarding budget, schedule, team composition, and other relevant project management aspects are identified and recorded.

② **Analysis**. The second step consists of gaining an understanding of the system that is to be gamified and its target users. Performing a *user analysis* starts with an identification of the potential target users and their segmentation into different groups. Afterwards, the needs and motivations of these individual user groups must be considered. These activities can be supported through player typologies adapted from video games (e. g., [Bar96], [HT14]), although the generalized applicability of such models is often challenged [Sch08, Dix11, Det15], and a context-specific, data-driven analysis of player personas is recommended as an alternative [Det15]. Complementing the user analysis, the *context analysis* consists of identifying, understanding, and documenting the business processes that should be supported through gamification, the corporate culture, and any technological constraints. Furthermore, the objectives formulated in the preparation phase should be operationalized by defining *success metrics* that can be used to evaluate the implementation.

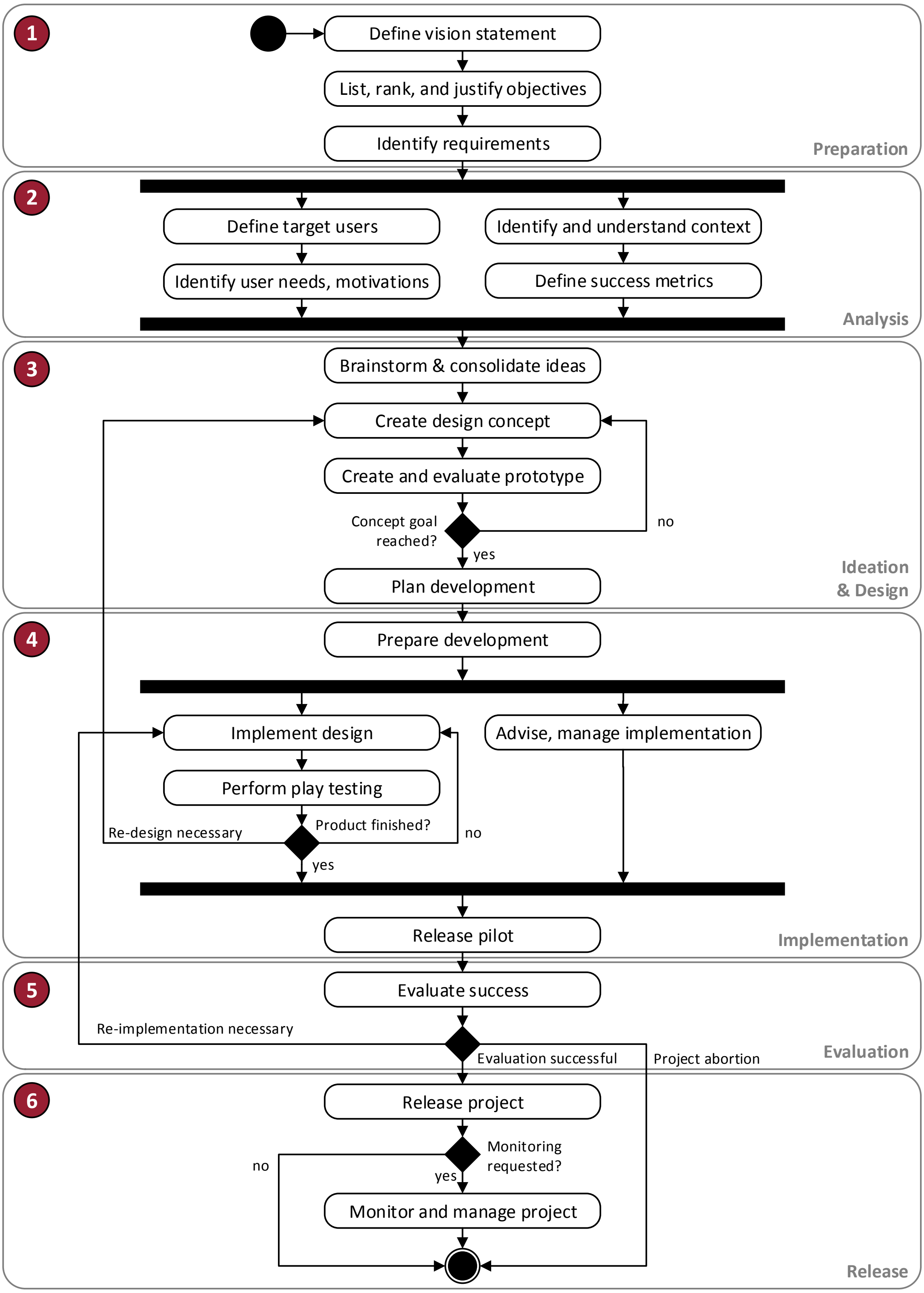


Figure 1.4: Gamification design method. Source: based on [MWHA17].

③ **Ideation and Design**. In the third step, a gamification design capable of satisfying the desired goals within the identified context must be developed. This task is subdivided into two broad activities: *ideation* and *design*. The former represents the creative, artistic side of gamification in which a large set of ideas is collected through brainstorming and is then consolidated. During ideation, a large variety of different approaches and tools can be used for inspiration, such as playing and discussing published board and video games, adapting patterns, principles, mechanics, heuristics, and other elements from game design practice (for a detailed discussion see Section 3.4), selecting from gamification best practices with proven effectiveness (see, e. g., [HKS14]), and employing design lenses to view the design space from multiple perspectives [Sch08, Det13, Det15]. In the design phase, the selected ideas are formalized as a concrete design concept, which is then tested through the creation and evaluation of playable, physical or digital *prototypes* [Ful08]. This step may be repeated iteratively until the test results of the prototype are satisfactory with regards to the previously defined objectives. Finally, the ideation and design phase is wrapped up by planning the development phase and documenting any information relevant for implementation as a development concept.

④ **Implementation**. The goal of the fourth step lies in creating a pilot release of the gamified application that can then be used for evaluation. It is initiated by the preparation of actual development, which particularly includes the decision whether the implementation should be created by an internal team or external developers. Furthermore, the use of existing gamification platforms and the potential impact of their technical constraints and limitations on the design concept should be considered. The development itself can be seen as a continuation of prototyping and is carried out as a cycle of alternating implementation and *play testing* activities that is repeated until the specified goals are achieved. This is supported by continuous project management to support, advise, and steer the implementation. Once the desired state of the gamified solution has been reached, it is finally released as a pilot. Should this prove to be impossible, the process may return to the design phase so that the concept can be adapted.

⑤ **Evaluation**. The fifth step is concerned with investigating whether the developed pilot is actually capable of meeting the objectives that have been defined in the preparation phase by measuring the success metrics established during analysis. Once again, a variety of quantitative and qualitative approaches can be employed for this activity, including interviews, surveys, controlled experiments, and play testing. Depending on the evaluation results, the next step may either be an abortion of the entire project if success seems to be unattainable, a return to the implementation phase if the product still needs to be adapted, or a continuation to the final step.

⑥ **Release**. In the sixth and final step, the gamification project is finally released to its target audience. The operation of the gamified product may be accompanied by post-launch *monitoring* in which the usage of the system is observed to compile a list of possible improvements.

A side-by-side comparison of this process and the DSRM process proposed by Peffers at al. reveals that gamification can—to a large extent—be interpreted as a specialization of design science. In particular, the correspondences between both methods are as follows: first, the activities carried out in the "preparation" and "analysis" phases of the gamification method and the "identify and motivate problem" and "define solution objectives" phases of the design science process correspond. Second, the activities "ideation and design" and "implementation" designated for gamification are reflected in the "design and develop artifact" phase of design science, although the former is defined with more detail and thus makes more concrete specifications about how these steps should be carried out. Third, although the "demonstration" activity as envisioned by the design science process does not have an explicit counter-part in the gamification method, the release of a pilot at the end of the "implementation" phase and its possible use in a field study prior to evaluation broadly corresponds to the former. Fourth, an "evaluation" activity exists in both, the gamification as well as the design science proposals. Furthermore, both methods allow for a return back to earlier phases in case of negative evaluation results, although the approach by Morschheuser et al. does not consider a re-specification of the solution objectives. Fifth, the "release" activity and the "communicate" phase of both processes broadly correspond with each other, although it may be reasonable to assume that the typical target audiences differ in both cases. While the research underlying this doctoral thesis was not conducted following a particular gamification method, its adherence to the DSRM process entails that it retrospectively also fits with the gamification design method by Morschheuser et al. due to the aforementioned correspondences. This is also reflected in the structure of this thesis as depicted in Figure 1.5.

1.4 Thesis Structure

The structure of this thesis is depicted in Figure 1.5. As illustrated, the remaining contents after the introduction are structured into three parts: the foundations in Part I, the application of gamification to BPM in Part II, and closing remarks in Part III. An appendix with supplementary information is contained in Part IV. Furthermore, most of the chapters within the main part directly correspond to individual activities of the applied research methods discussed in the previous section. Lastly, some of the academic papers (co-)written by the author of this thesis also contribute to specific chapters.

Part I: Foundations. In the first part, the theoretical foundations on which the remainder of this thesis is built are laid out. The first of these areas is Business Process Management which is discussed in Chapter 2. By following a bottom-up approach, this chapter elaborates on essential terms of this research field, including *business process*, *business process model*, *business process modeling*, and *business process management*. Then, the topic of *business process model quality* is addressed by discussing it on a conceptual level and then providing concrete examples of quality metrics, guidelines, and frameworks that are employed within the BPM domain. The second main area is gamification, which is discussed in Chapter 3. After presenting a brief *history* of the concept, the term gamification is *defined* and *related fields* are outlined. Next, a number of prominent *examples* from gamification practice as well as academia are introduced to make this approach more tangible. This is followed by a systematic, multi-level examination of the *game design elements* that may be used in gamification while contrasting theory with practice as reported in relevant literature reviews. Subsequently, an overview of the most important *theories* (most notably from behavioral psychology) frequently used to explain how

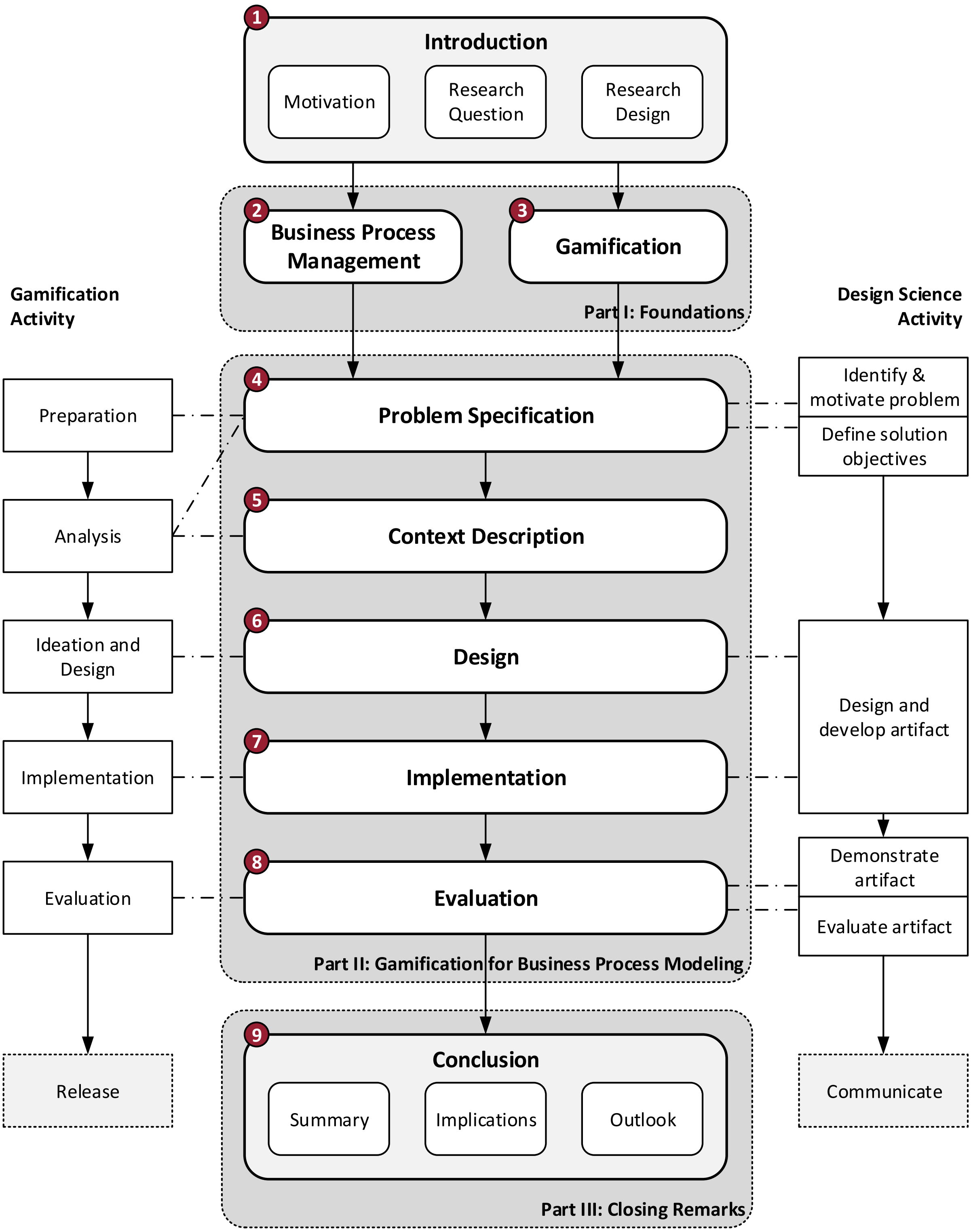


Figure 1.5: Structure of this thesis.

and why gamification works is provided. The chapter closes with an overview of the most important points of *criticism* that gamification is faced with. This part of the thesis is written in a self-contained fashion that is independent of the subsequent application part. Consequently, it provides a comprehensive, state-of-the-art overview of relevant aspects of BPM and gamification that includes, but also exceeds what is required for the understanding of part II.

Part II: Gamification for Business Process Modeling. In the second part, the research question established in Section 1.2 is examined through the conceptualization, implementation, and evaluation of an IT artifact—specifically an instantiation—that integrates gamification functionality into a business process modeling tool. To that extent, Chapter 4 first discusses recent trends in BPM research—in particular Social BPM—that point to an increasing interest in issues related to the human individuals BPM involves as compared to a historic focus on methods and IT. The chapter further examines the challenges resulting from more human-oriented BPM and identifies the potential contributions that gamification can make to overcome them. From the latter, concrete goals that the artifact to be created should achieve are derived. Having identified the need for research, Chapter 5 then continues to describe the context in which this research project was carried out, most importantly the Horus Method and the Horus Business Modeler. Afterwards, Chapter 6 provides a detailed description of a design concept for the integration of gamification into a business process modeling tool. This concept draws inspiration from various sources, including commercial games and gamification best practices, and is subdivided into groups of similar features. Its implementation and the specific challenges that occurred in this context are then presented in Chapter 7. The second part ends with an evaluation of the implemented concept in Chapter 8 based on data gained from a field experiment and a lab experiment.

Part III: Closing Remarks. In the final part, this thesis concludes with a brief summarization of its most important contents, a discussion of the implications that its findings may have for research and practice, and an outlook for future work that can be based on the former.

Research method correspondence. As illustrated in Figure 1.5, the individual chapters of Part II mirror the activities of the applied DSRM process and gamification method. With the exception of the context description in

Chapter 5, each chapter is reflected in both processes, albeit not in a symmetric fashion. The most notable differences are as follows: firstly, the definition of success metrics which is part of the gamification "analysis" phase is contained in Chapter 4. Secondly, both of the first two design science phases are also addressed by Chapter 4. Thirdly, whereas the gamification activities "design" and "implementation" have direct correspondences in Chapter 6 and Chapter 7, they are summarized as a single step in the DSRM process. Lastly, even though design science consists of separate "demonstration" and "evaluation" phases, both are addressed together in Chapter 8.

Supplementary papers. Five academic papers supporting and supplementing the contents of various chapters have been written over the course of this research project. Firstly, a literature review discussing the current state of Social BPM as a holistic approach to human-centric BPM as well as the challenges that occur in such a setting are provided in [PV13b] and [PV14]. Secondly, the European Research Center for Information Systems (ERCIS) working paper [PBSV15b] as well as the journal publication [PBSV15a] expanding upon the former introduce a concept based on a business process modeling case study for the lecture "Introduction to IS" that has served as the foundation for the demonstration and evaluation of the developed IT artifact. Lastly, the paper [Pfl16] was presented at a PhD workshop, and thus gives a high-level overview of the entire research project.

Part I
Foundations

2 Business Process Management

The first of the two pillars on which the main part of this thesis is built is Business Process Management (BPM). BPM is an extensive field concerned with a variety of different topics that affect the business processes of an organization, including strategic alignment, governance, methods, information technology, people, and culture [RvB10]. Thus, the main purpose of this chapter lies in giving an overview of those areas that are relevant in the context of this work. To that extent, Section 2.1 first defines the term *business process* and illustrates what types of business processes can be found in organizations. Afterwards, Section 2.2 specifies what a *model* of a business process is, outlines for which purposes such models are commonly used, and contrasts different types of model representation. Next, the process of *business process modeling* and the various components that it involves are discussed in Section 2.3. Finally, Section 2.4 gives a brief overview of the entire *BPM discipline* and describes the role that process modeling has in the former. This concludes the first logical part of this section, which is focused on general business process foundations. The remaining three sections are application-specific and thus present more concrete information that is operationalized in Part II. Firstly, Section 2.5 defines a *canonical representation* for business process models based on general graphs that is independent of the underlying modeling language. Then, Section 2.6 characterizes the notion of *quality* for business process models and illustrates the role of quality metrics, quality frameworks, and quality guidelines for the former. Finally, an in-depth discussion of different types of process model *quality metrics* is presented in Section 2.7, thus closing this foundational chapter.

2.1 Business Processes

Generally speaking, the term *process* refers to a "continuous and regular action or succession of actions occurring or performed in a definite manner, and having a particular result or outcome" [26]. Following this definition, the activity of walking can be understood as a very simple process consisting of the repeated actions "put left foot in front of right foot" and "put right foot in front of left foot", with the outcome being the displacement of the walker. If the actions of a process are performed by an organization with the purpose of fulfilling the needs of its customers, it is called a *business process* [BKR03]. Examples for business processes include the credit application and approval process in banks, the claim settlement process in insurances, and the car manufacturing process in the automotive industry. As this thesis is only concerned with business processes, any subsequent use of the term "process" alone will implicitly assume the existence of an organizational context.

Looking at specific definitions of the term *business process*, work in the area of BPM provides a plethora of different proposals, most of which are conceptually very similar. For this reason, there is currently no single definition that is universally agreed upon. Some of the most widely-used definitions are the following. In 1993, Hammer and Champy characterized the term as "a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer" [HC93]. Also in 1993, Davenport defined business processes as "a specific ordering of work activities across time and place with a beginning, an end, and clearly defined inputs and outputs" [Dav93, p. 5]. A third definition was proposed by Becker et al., who consider a business process "a completely closed, timely and logical sequence of activities which are required to work on a process-oriented business object" [BKR03]. Lastly, Weske defines the term as "a set of activities that are performed in coordination in an organizational and technical environment [...] to jointly realize a business goal" [Wes07]. By consolidating these various definitions, the following characteristics of a business process can be identified:

- **Boundaries**. A business process has clearly defined starting and ending points that delineate it from other processes. At its boundaries, the required inputs and the outputs that it yields are specified. For instance, the process of manufacturing a car *to order* starts with a production order and ends with the completed product. At its boundaries, it is connected to preceding processes such as order processing and succeeding processes such as outbound logistics. Furthermore, the manufacturing process itself may be divisible into numerous subprocesses with respective internal boundaries.

- **Ordering**. A business process consists of a set of activities that are executed with a specific ordering. This order is clearly defined in terms of the logical and chronological interrelationships between activities. The order of activities may not only be purely sequential, but can also exhibit choices between alternative sets of activities, parallel activities, or iterations. For instance, the order in which the individual steps of manufacturing a car must be carried out are determined by the layout of the assembly line. However, certain components, such as the body and the engine, can be constructed in parallel until they are joined, at which point the process must be synchronized.

- **Value creation**. By consuming inputs to create outputs, a business process creates value. In particular, the outputs directly or indirectly contribute to the realization of the business goals of an organization and create value for its customers. For instance, the inputs that are consumed by the car manufacturing process include raw materials and components, whereas the output is a finished car with a certain configuration. Clearly, the outcome of this process has a higher value than the sum of its individual components, and thus value was created. For the manufacturer, this value manifests itself as revenue, whereas the value for the customer is a car that meets their needs.

- **Environment**. A business process does not exist in a vacuum, but is embedded in an organizational and technical environment, in which its execution is carried out in a coordinated fashion. For instance, the organizational environment of the car manufacturing process includes marketing, sales, and logistics, whereas its technical environments includes machines and software systems that are used.

Business processes exist at all levels and in all areas of an enterprise, and due to the breadth inherent to all definitions of the term, all work activities carried out in organizations can essentially be considered to be parts of business processes. However, many enterprises instead focus on the hierarchic organization of *who* does the work rather than *how* work should be done, i. e., they assume a function-oriented rather than a process-oriented perspective [Dav93]. This causes the emergence of functional silos and highly fragmented business processes with uncoordinated handoffs between the functions of different departments [HC93]. Possible consequences include inadequate process performance, an increased difficulty of improving or innovating processes, and unresponsiveness to changes in the external environment [Dav93, HC93]. Thus, it is often advocated that enterprises should acknowledge the boundary-spanning, cross-functional, and cross-organizational nature of numerous work activities by orienting their business practices around business processes (i. e., horizontally) rather than functional hierarchies (i. e., vertically) [HC93, Dav93, BKR03].

Looking at the different processes within an organization, it is intuitively clear that there are many different types of processes with varying characteristics. For instance, while some processes may make a larger contribution to its overall value creation, other processes only provide support for the former. Overall, while today there is no single accepted taxonomy of process types, some common classifications are as follows:

- **Repetitiveness vs. value** One of the most important taxonomies (originally proposed for workflows in [McC92]) distinguishes between four types of processes along the dimensions *repetitiveness* (or predictability) and *value* (or criticality) [AAAM97, GT98]. *Ad hoc* processes are characterized by a low repetitiveness and low value. Each instance of such a process is highly unique such that there is no predefined pattern for carrying it out and a high degree of human involvement and decision-making is required. *Collaborative processes* are similarly unique, but are mission critical for an enterprise and thus of high value. Compared to an ad-hoc process, a collaborative process may involve more iterations over its activities and a larger number of people. The other end of the spectrum is formed by processes with high repetitiveness. Here, *administrative processes* are characterized by low value and generally represent bureaucratic activities with clearly defined steps and rules. *Production processes* are also highly structured, but represent that type of work that is the most critical for a firm and contributes the largest part to its value creation. Such processes often have a larger scale and complexity and are carried out in a heterogeneous environment involving multiple parties and information systems.

- **Value chain**. Another classification of business processes is based on Porter's value chain [Por85, PM85] and distinguishes between processes that directly contribute to the value creation of an organization and processes that only do so indirectly. The former are typically called *primary processes* [RB13b], *core processes* [Oul95], *operational processes* [Dav93], or *main processes* [vRKH15]. Examples for this type of process can be found in the areas of inbound and outbound logistics, operations, marketing and sales, and customer service. Non-value creating processes are commonly subdivided into two additional categories [Dav93, Oul95, RB13b, vRKH15]: *support processes* and *management processes* (although Davenport does not make such a distinction and only mentions the latter type). Support processes create outputs that are invisible to the customer, but are still required for effective business operation. They are related to, e. g., to firm infrastructure, human resource management, technology management, and procurement. Lastly, management processes relate to activities carried out by managers and address factors such as corporate strategy, resource allocation, and goal-setting.

- **Physicality**. Some authors distinguish between processes whose consequences manifest themselves in the physical world and those whose consequences do not, which are called material and information processes respectively [MMWFF92, GHS95]. A *material process* is rooted in the "physical world" and as such relates to the assembly of physical components by human activity or machines, ultimately resulting in the delivery of a physical good. Conversely, the focus of *information processes* is on automated or semi-automated tasks that manage information in some way. Thus, they make use of the information systems of an organization and rely, e. g., on technologies for databases, distributed systems, and transaction processing.

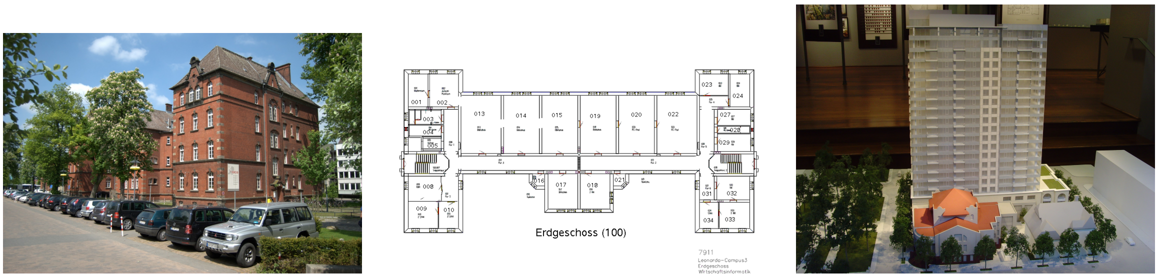
- **Degree of automation**. Finally, business processes can be distinguished according to their potential to be automated through the use of IT,

> which is in fact one possibility how organizational investments in IT can contribute to the creation of competitive advantage [PM85]. From this perspective, two types of processes can be distinguished [GHS95]: *human-oriented* processes in which tasks are completed through the collaboration and coordination of humans, and *system-oriented* processes that involve a high degree of computational tasks that can be performed in an automated fashion through communicating software systems.

As it can be seen, these classifications and taxonomies are not mutually exclusive. For instance, administrative processes are related to supporting processes, production processes can roughly be equated to operational processes, and system-oriented processes will often be information processes as well. Furthermore, some process types are related to other concepts that are similar to business processes, yet conceptually different. For instance, *workflows* represent a special kind of business processes whose objective is (partially) automated process execution through the support of software systems called Workflow Management Systems (WfMSs) [GHS95, BKR03]. Thus, according to Becker et al., a workflow is "that part of the process that contains the timely and logical sequence of activities of a job as well as information about the data and resources that are involved in the execution of this job" [BKR03, p. 264]. Another related concept are *cases*, which represent very flexible and knowledge-intensive business processes for which each real-world instantiation may potentially take a different course, such as medical treatments [vdAWG05]. In contrast to "normal" business processes, the set of tasks that can be executed for a concrete case at a given point in time does not depend on a pre-defined ordering of activities, but on the current state and structure of the case. Thus, the main role of software systems in this context lies in providing knowledge workers with all required information that enables them to decide what to do next to achieve the business goals of the case.

2.2 Business Process Models

To communicate about business processes, they must be made explicit in some way, for instance using a natural language. Indeed, one of the earliest known descriptions of a process was recorded in English in 1776, when Adam Smith



(a) Photograph of a building. **(b)** Floor plan of a building. **(c)** Physical architectural model.

Figure 2.1: Sample types of models.

outlined the pin manufacturing process of a fictional pin factory [Smi76]. In formal terms, such a description is called a *model*: a *representation* of a real or artificial original which may itself be a model as well. In general, such a representation is an *abstraction* of the original, i. e., it does not include all attributes of the latter, but only those that are deemed relevant by its creators and/or users. Finally, meaning is given to the model by its users at a given time for a given purpose, i. e., it depends on the context in which the model is used and thus involves *pragmatics* [Sta73].

These three characteristics are exemplified by Figure 2.1. Firstly, Figure 2.1a presents a photograph of a building and part of its surroundings. While reproducing the visual elements of the building in a realistic fashion, the picture in itself clearly is a model as it abstracts from some properties of the building such as its interiors. More interestingly, Figure 2.1b shows a different representation of the ground floor of the same building. In this illustration, rooms are depicted to scale, room numbers are displayed, and doors and stairwells can be seen. Clearly, this model abstracts from numerous real-world properties, such as colors and ceiling heights. Furthermore, it may serve different purposes in different contexts. For instance, one model user might use it to plan furnishing the building, while another person could use it for key management purposes. Lastly, an architectural model of another building is shown in Figure 2.1c. This may for instance be a representation of a planned structure, and the purpose of the model could then be to convince investors to provide the necessary funds to build it.

Analogously, a *business process model* is a representation of a real or theoretical business process that abstracts from some of its details and that is

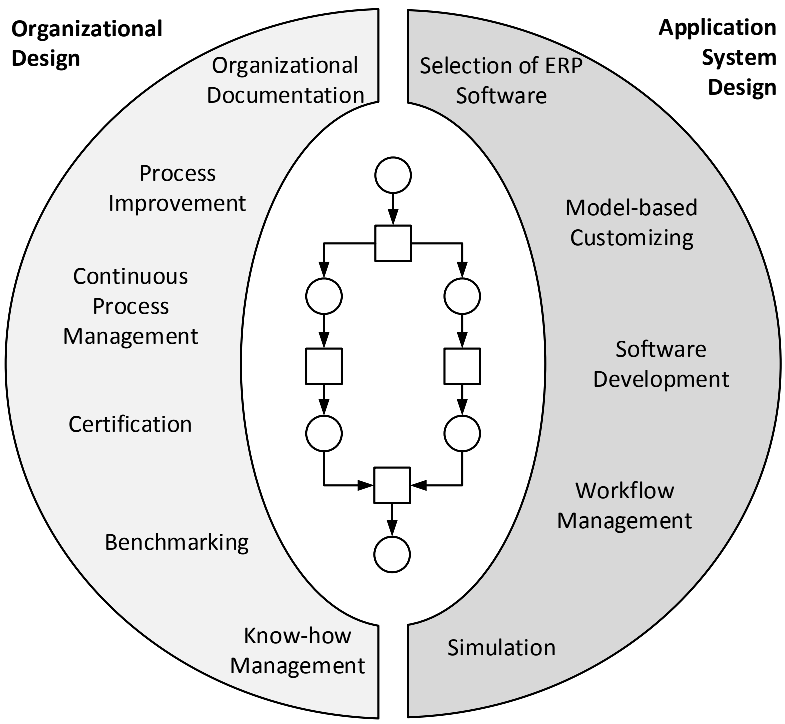


Figure 2.2: Purposes of process models. Source: based on [BKR03, BPV12].

used within a particular context for a certain purpose. These properties also manifest themselves in the hypothetical pin manufacturing process mentioned at the beginning of this section, which abstracts from certain details (e. g., the tool used for cutting the wire) and, at the time, served the purpose of discussing the impacts of labor division [Smi76]. Generally speaking, a business process model can be used for a wide variety of different *purposes* and, depending on its function, may need to contain different kinds of information. For instance, Ould proposes that business process models can be used for the description, the analysis, and the enactment of business processes [Oul95]. Rosemann instead distinguishes between using process models for *organizational design* and for *application system design* as shown in Figure 2.2 [Ros03]. Whereas the former is concerned with the use of models to document, study, and improve the design of business practices, the latter addresses the selection, design, and analysis of application systems that support business processes. Some of the most important examples for how business process models may contribute to these two main purposes are as follows [Ros03]:

- **Organizational documentation**. Companies can use process models to maintain a complete and up-to-date documentation of their as-is business processes, although few do so in practice. This is for the purpose of making processes more transparent and being able to efficiently communicate them to employees.
- **Process improvement**. Organizations are often interested in using business process models for the purpose of process reorganization and improvement. This requires models that unveil the weaknesses of current business practices. A certain degree of formalism is required so that models of as-is and to-be processes can be compared.
- **Continuous process management**. Subsequent to reorganization, business process models can enable planning, executing, and controlling processes in the long term. This requires comparing process models to actual process execution and investigating any deviations, which may either be the result of an inadequate model or deficiencies in process enactment.
- **Certification**. To obtain an ISO 9000 quality management certification, companies must be able to provide a high-quality documentation of their quality assurance processes. This requires, among other artifacts, business process models, a documentation of model changes, and proof that processes are executed as specified in their models.
- **Benchmarking**. Business process models can be used to enable a comparison of process structure and process performance with internal or external references that should represent better or best practices. This requires models that contain relevant benchmarking information as well as the availability (and also comparability) of information about benchmark processes.
- **Know-how management**. To increase the transparency of corporate knowledge, process models can be used to support the acquisition and transferal of know-how. For that purpose, special types of process models having "knowledge" as process inputs and outputs and links to the organization structure are required.

- **Selection of ERP software**. Enterprise Resource Planning (ERP) tools provide comprehensive solutions for many core and support processes of an organization. They often document their functionality with reference process models, which can be compared with own process models to determine the coverage of tool requirements. Major challenges here arise from differences in naming, layout, and other conventions between both model sets.

- **Model-based customization**. Building on the previous purpose, reference models can also be used to determine how an off-the-shelf application system must be customized to be able to fulfill company requirements.

- **Software development**. A common use-case for models lies in the documentation of the requirements for a software that is to be developed. Here, process models can also play an important role by describing the usage procedures the finished product should enable. Models for that purpose are typically rather formal and contain references to data models and other model types relevant for software development.

- **Workflow management**. As previously noticed, a workflow is a special type of process with a focus on semi-automated process enactment through the support of a WfMS. For that purpose, business process models must contain detailed information about the roles, data, and application systems involved in the execution of the process. Typically, such models also have a higher level of detail than, for instance, models that are used for organizational documentation.

- **Simulation**. Business process models may be used to simulate the execution of a process over time to measure its performance, detect weaknesses, compare different process scenarios, and calculate the required number of resources such as humans. Models used for that purpose must contain detailed information about factors such as costs and processing times of activities and probabilities at decision points.

In the example used at the beginning of this section, Smith used the English language to create a textual representation of a fictional pin manufacturing

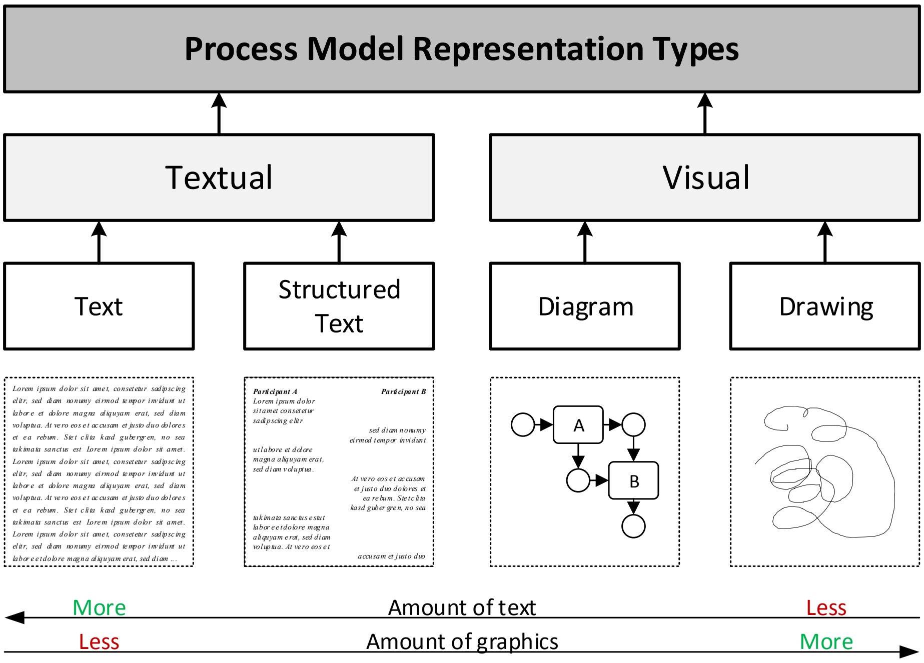


Figure 2.3: Representation types for process models.

process [Smi76]. Besides such a purely textual form, business process models can also be created using visual elements [Moo09]. However, the two are not mutually exclusive and between "only text" and "only graphics" a wide range of mixed representations is possible [RSR10, RSR12]. Figure 2.3 illustrates some of the most common representation types for business process models (cf., [GPR95, RSR10, RSR12, FR16a]):

- **Text**. A textual representation of a business process describes its individual activities and their interrelationships using a natural language. Possible sources for textual process descriptions in organizations are manifold and may include, for instance, reports, manuals, knowledge management systems, and emails [FMP11]. Today, organizations produce, store, and analyze enormous amounts of data that grows at faster and faster rates [BBFRS12]. This is especially true for unstructured data such as texts, which represents the majority of the data that is available [BA03, BBFRS12, KAEM13]. Thus, texts have been noted as an important source for business process descriptions with large potential value for various uses [FMP11].

- **Structured text**. A structured text representation of a business process has condensed textual contents (e. g., no stop words, no full sentences) but is enriched with structural elements such as indentation, text blocks, or specific keywords [VW86, FR16a]. This way, process descriptions gain a pseudo-algorithmic nature and thus sit in the middle between natural language and pseudo code as used in programming contexts.

- **Diagram**. A diagram is a representation of a business process in two-dimensional space [LS87] that uses a visual notation with clearly-defined rules and symbols to express the activities, objects, dependencies, and other aspects of the process [Moo09]. Usually, business process diagrams consist of graphical shapes such as rectangles or circles and lines with or without arrowheads connecting them to specify interrelationships. Textual elements still play an important role in this representation type as they specify the actual semantics of what is depicted [Leo13]. Recker et al. also describe a "hybrid" design type which combines diagrams with concrete graphics for certain actors and objects to enhance the former [RSR10], which is argued to make models more understandable and visually appealing [Pet95, MPG08, Moo09, MRR10a]. A comprehensive, general review of research on diagrams can be found in [Pur14].

- **Drawing**. A drawing strongly relies on concrete graphical components instead of abstract shapes to depict a business process and does not need to follow any clearly-defined rules. Depending on the use of abstract graphics and textual elements, Recker et al. distinguish two different types of process drawings [RSR10]. Firstly, *storyboards* utilize some abstract graphics such as lines and rectangles and may also contain small amounts of text. Such a representation may also be referred to as a "rich picture" [AGS92]. Secondly, models following the *canvas* design type aim to illustrate the entire process using only concrete graphics and as such only make negligible use of other components. Whereas the former can still represent some notion of order and dependency between work activities, this is not the case for the latter.

The question which of these types of representations is superior does not have a definitive answer. However, discussions about business process models in academia most commonly assume a diagram-based representation using one of several widely-established graphical notations. This may be due to the reason that many scholars attribute certain advantages to visual representations. For instance, Nordsieck points out that visualizations using geometric symbols may not only serve as a complement or replacement for natural language, but often illustrate the essence of the depicted subject matter in a much more comprehensive manner than any other type of representation can [Nor50]. Furthermore, he points out that a (good) diagram is characterized by conciseness, clarity, and accuracy, and that its pictorial nature and use of two-dimensional space reveals the interrelationships between model elements and facilitates the analysis of structures and procedures in organizations. Indeed, the idiom “a picture is worth a thousand words” indicates that the superiority of visualizations is accepted at least for some purposes and contexts [LS87]. However, this is not to say that business process models should *only* consist of graphics and avoid the use of text entirely, as *dual coding theory* [Pai86] suggests that humans possess separate systems to process verbal and visual stimuli, and thus both should be used to supplement each other [Moo09]. Eventually, the choice between textual and visual representations may be less clear when taking into account the personal characteristics of model users, as there are indications that modeling experts can work with visual business process models more effectively and efficiently [Pet06, OFR+12, MSR12]

Despite the dominant role and superiority of diagram-based notations, a case must be made for the use of text in BPM. Due to the wealth of unstructured data that is readily available in organizations today, texts as a source of valuable information about business processes should not be discounted. However, interpreting this data is difficult due to its inherent ambiguity, especially in comparison to structured data, thus making it challenging to realize value from it [BBFRS12]. As a consequence, textual model representations remain largely unstudied in Business Process Management (BPM) so far despite the large potential value they might hold [FMP11]. Nevertheless, some research in this area exists and addresses, for instance, the generation of business process diagrams from process descriptions in natural language [FMP11, SMSZ13, RTT16] and vice versa [LMP12, LMP14], validating business process diagrams against textual process specifications [LMP14, vdALR15], and generating appropriate names for models or model fragments [LMR11, LMRL14].

2.3 Business Process Modeling

Business process modeling is the "human activity of creating a business process model" [Men08, p. 11]. The person who performs this act constructs a representation of a business process for a specific purpose based on their perception of the former and thus becomes known as a *process modeler* [BPV12]. To that extent, process modelers make statements about business processes using a language, which may for instance be a natural language such as English, or a *process modeling language* specifically designed for that purpose, such as Event-driven Process Chains (EPCs) [Sch00], the Business Process Model and Notation (BPMN) [CT12], or particular types of high-level Petri nets [Jen97], such as WorkFlow nets [vdA98] or Extensible Markup Language (XML) nets [Len03]. It should be noted that despite the characterization of process modeling as a human activity, process models may sometimes be created by computers, for instance in the context of process mining [vdARS05, vdA11].

Business process modeling been characterized as a very time-consuming endeavor requiring costly investments in software tools, licensing fees, methodologies, and employee training [IGRR09, FMP11, BR12]. Therefore, a clear and convincing value proposition is required to obtain executive management support and the required funds to carry out modeling projects [IGRR09, IRRG09]. Based on a review of relevant literature, Indulska et al. suggest that the benefits of business process modeling can fall into five general categories [IGRR09]: strategic benefits, organizational benefits, managerial benefits, operational benefits, and IT infrastructure benefits [IGRR09]. Furthermore, through a series of interviews, Bernhard and Recker identified the following, more specific process modeling impacts: first, it facilitates process understanding and allows individuals to develop process knowledge. Second, it enables communication about business processes in a team and supports coordination. Third, it allows identifying improvement potentials for business processes and thus facilitates decision-making. Fourth, it increases the satisfaction of working on business processes. Fifth, it facilitates achieving process improvement goals more effectively and efficiently. Nevertheless, there is not enough research providing empirical evidence for the benefits of process modeling yet, and thus this issue has been identified as an important future challenge for BPM research by academics, practitioners, and tool vendors [IRRG09].

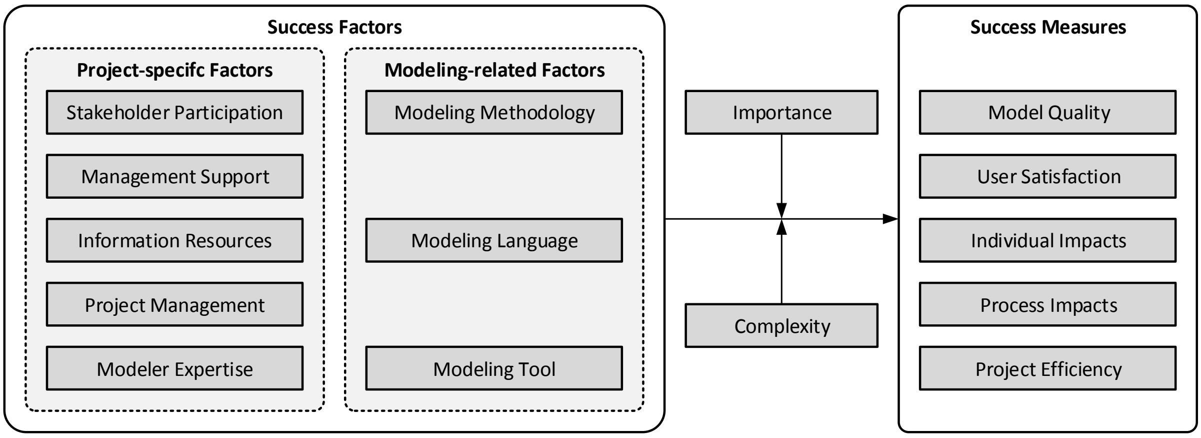


Figure 2.4: Success factors of business process modeling. Source: based on [BGR05, p. 357].

As modeling projects may have far-reaching organizational consequences such as the implementation of new business processes and organization structures, it is important that the success of such initiatives can be measured [BGR05]. Generally speaking, a modeling endeavor can be said to be successful if it is both *effective* (i. e., the objectives relating to its purpose are achieved) and *efficient* (i. e., it adheres to the allocated time and budget). More specifically, the success of a process modeling project depends on a variety of *success factors* and can be evaluated through different *success measures* as illustrated in Figure 2.4. In total, Bandara et al. determined eight different influence factors on modeling success that either relate to the concrete modeling project or the employed modeling tools and techniques, and five indicators through which modeling success can be measured. The relationship between these constructs is mediated by the complexity of the relevant processes and the importance of the modeling project. For more detailed information, the reader is referred to [BGR05]. Additional factors that can influence the success (or failure) of process modeling endeavors can be found in [Ros06a] and [Ros06b].

Modeling in general and business process modeling in particular involves a set of different, interrelated components as depicted in Figure 2.5. This illustration is based on the model presented by Karagiannis and Kühn [KK02, p. 3] as adapted by [Men08, p. 8] and [Leo13, p. 12], and the conceptual modeling research framework proposed by Wand and Weber [WW02a, p. 364]. One of the main building blocks is the *modeling technique*, which in turn consists of two

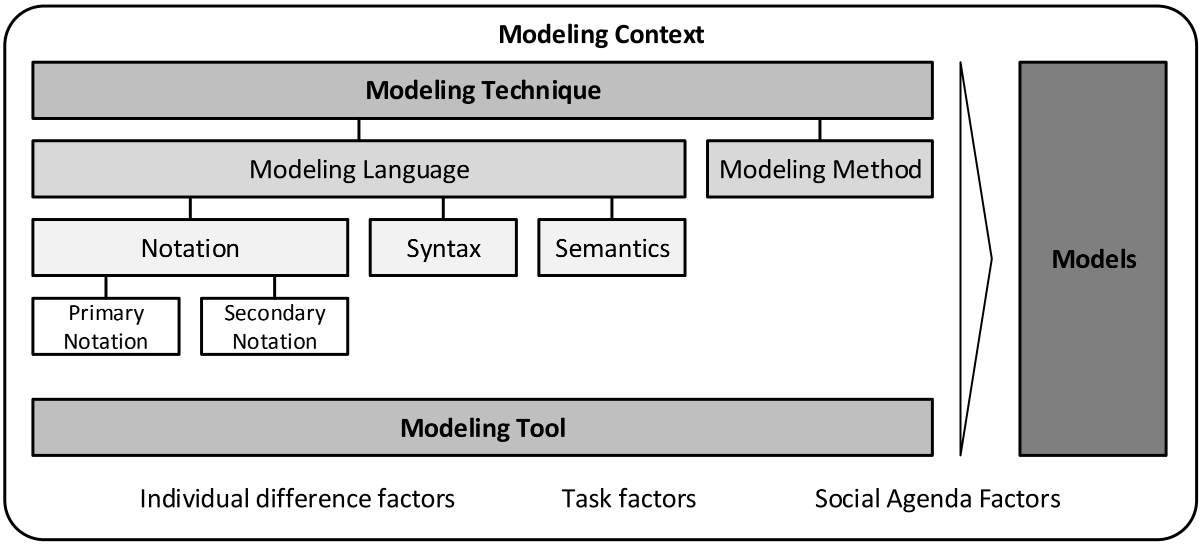


Figure 2.5: Components of business process modeling.

parts: the *modeling language* and the *modeling method*. Whereas the language defines the *syntax*, *semantics*, and *notation* for creating valid models, the latter prescribes practices for how the former should be used. When used together with a *modeling tool*, the modeling technique enables the creation of actual *models*. Lastly, it is important to note that modeling always takes place within a particular *modeling context*. As the terminology used in modeling literature is highly heterogeneous, alternative terms for the components presented here can commonly be found, e. g., "modeling grammar" for "modeling method" [WW02a], "modeling procedure" for "modeling method" [KK02], or "modeling script" for "model" [WW02a]. In the following paragraphs, the components presented in Figure 2.5 are discussed in more detail.

Modeling Language

Starting with the invention of Petri nets by Carl Adam Petri in 1962 [Pet62], a plethora of different modeling languages have been developed or adapted for business process modeling over the course of the last five decades. This is reflected by an examination of relevant survey papers, which contrast and compare as many as 6 [Gia01], 7 [LK06], 12 [AS04, RRIG09], and 19 [MTJ+10] modeling languages, respectively. Whereas some of these languages are especially well-suited for pure process description (Event-driven Process Chain (EPC), Business Process Model and Notation (BPMN)), other languages may be more appropriate for simulation (Petri nets) or automated process execution (Business Process Execution Language (BPEL)). Furthermore, modeling languages may also differ in their capabilities to represent various business process constructs [LK06, RRIG09]. Considering all existing process modeling languages is not a viable approach for the remainder of this thesis, and thus a representative language must be selected for any further deliberations. As the Horus Business Modeler, on which the implementation described in Part II is based, utilizes an extended version of Petri nets as one of its core modeling languages, the choice naturally falls on Petri nets. However, the definition of a language-independent process representation in Section 2.5 ensures that findings presented in the following can also be applied to other process modeling languages. Due to the extent of the body of knowledge on Petri nets, only the fundamental basics of this modeling language can be addressed in the following paragraphs. For more detailed information, the readers is referred to one of the many textbooks and surveys on the topic [Rei82, RW82, Mur89, RR98a, RR98b, GV03, vdAS11].

Syntax. The syntax of a process modeling language defines the set of constructs that it provides and specifies rules regarding their possible combination [Men08]. For instance, a basic Petri net is comprised of a set S of *places*, a set T of *transitions*, and a set F of *arcs* (also called *flow relation*). Using these sets, a Petri net can formally be defined as

$$N = (S, T, F) \tag{2.1}$$

where S and T are finite, disjoint sets, i. e., $P \cap T = \varnothing$, and arcs may only connect places with transitions and vice versa, i. e. [RR98a],

$$F \subseteq (S \times T) \cup (T \times S) \tag{2.2}$$

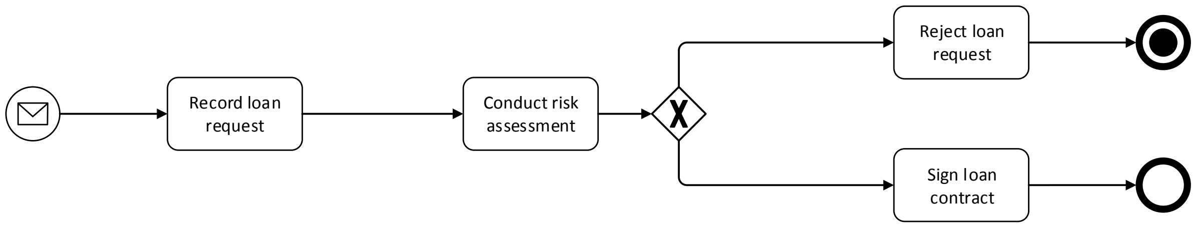
From the perspective of a single transition $t \in T$, any place $s \in S$ for which an arc $(s, t) \in F$ exists is called an *input place*, whereas any place for which an arc $(t, s) \in F$ exists is called an *output place* [vdA98]. The set of input places is also called the *preset*, and the set of output places the *postset* of t. Further, a *marking* of a Petri net is a function $m : S \to \mathbb{N}$ that assigns each place zero or more *tokens*. Together, tokens represent the *state* of a Petri net and influence its possible behavior as described in the next paragraph. As Petri nets were not originally created as a modeling language for business processes, this terminology does not directly reflect the concepts used in the definitions presented in Section 2.1. However, transitions can be seen as activities, computation steps or events, input places as preconditions, input data, or resources needed for the execution of the former, and output places as post-conditions, output data, or released resources [MTJ+10]. The constructs found in other process modeling languages include activities, events, gateway, artifacts, and connections for BPMN [CT12], and functions, events, arcs, and connectors for EPC [Sch00, BKR03].

Semantics. The semantics of a process modeling language "bind the constructs defined in [its] syntax to a meaning" [Men08, p. 9]. Petri nets have formally defined execution semantics that are based on tokens and *firing rules* [vdA98]. Specifically, a transition is said to be *enabled* if each place in its preset contains at least one token. If a transition is enabled, it may *fire*, thereby *consuming* one token from each input place and *producing* one token in each output place. In conjunction with the interpretation of transitions and places mentioned above, an enabled transition can be seen as a process activity which fulfills all preconditions for execution, and firing it equals carrying the activity out. As transitions successively fire, the state of a Petri net is modified and thus its marking changes. Once again, this can be compared to the state of a business process instance changing over time. Finally, each Petri net also has an *initial marking* m_0 that has a direct influence on the possible states it may reach [RR98a]. Since their inception, the syntax and semantics of Petri nets have been modified through numerous proposals. This has allowed extending the modeling language with concepts such as individual, distinguishable tokens that can carry data [Jen91], Boolean expressions for transitions and arcs that restrict when transitions can fire based on token data [Jen91], places with restricted token capacities [CH93], the hierarchical subdivision of models into multiple, interrelated sub-models [HJS91], and time required by transitions for firing [Jen97]. Furthermore, specializations for business process modeling were developed, for instance WorkFlow nets [vdA98] and XML nets [LO01, Len03]. The fact that Petri nets have precise semantics despite their extensive capabilities is often described as one of the major advantages of this modeling language that eliminates the possibility for any ambiguities regarding process enactment [vdA98]. However, considerable work has also gone into formalizing the semantics of other modeling languages such as

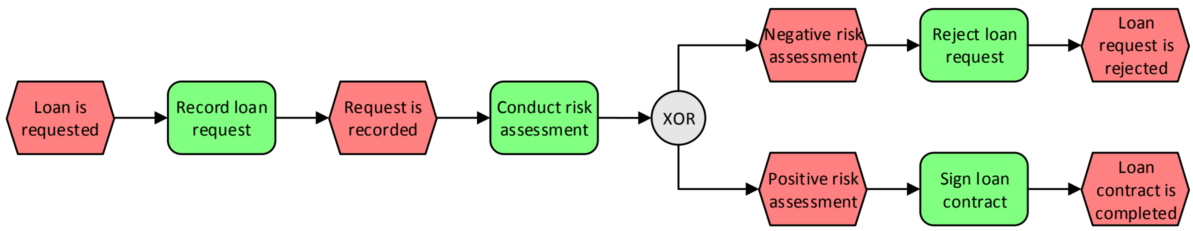
EPC [vdA99, Men08, ch. 2.4] and BPMN [DDO08, VD13], ironically often by mapping the latter to Petri nets.

Primary notation. The primary notation of a process modeling language describes how its syntactical constructs are visualized [KK02]. Generally speaking, this may be done via any of the representation types discussed in Section 2.2. However, most common languages used in the context of BPM are diagram-based and thus define specific graphical shapes for their elements. For instance, Petri nets are visualized by drawing transitions as rectangles, places as circles, arcs as directed arrows, and tokens as filled circles [vdA98]. In contrast, the notation of EPCs requires events to be drawn as purple hexagons, functions as green, rounded rectangles, arcs as directed edges, and connectors as circles with a particular symbol depending on connector type [Sch00]. Lastly, BPMN diagrams are drawn by activities as rounded rectangles, events as circles, gateways as diamonds, and connections as directed arrows [CT12]. These three notations are contrasted in Figure 2.6, which depicts a simple loan request process using all three modeling languages. It should be noted that a process modeling language may also have more than one primary notation. For instance, while Petri nets are a graphical modeling language, its mathematical syntax also allows defining models as structured texts by specifying sets of transitions, places, arcs, and markings. Thus, there are at least two valid notations for representing Petri nets.

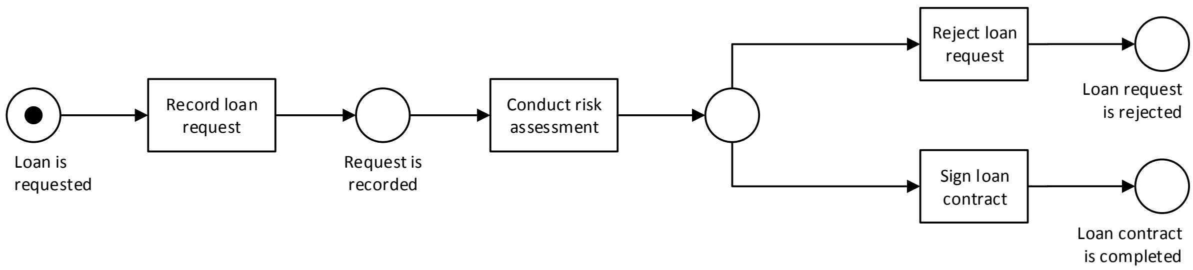
Secondary notation. Despite the restrictions imposed by the primary notation of any modeling language, there still exists an infinite way to visually represent any business process model by finding a configuration of graphical properties such as position, size, color, and texture. This is enabled by the secondary notation of a modeling language, which consists of “things which are not formally part of a notation which are nevertheless used to interpret it” [Pet06, p. 293]. An example for this is the convention that the elements of a process model are usually positioned in a way so that the directions of arcs point from left to right (or sometimes from top to bottom) [FS14]. While no widely-used modeling notation makes any prescriptions to that extent, this convention has a high acceptance nevertheless due to its compatibility with the text reading direction in most Western cultures. Clearly, secondary notation is not limited to process modeling, but is also widely-used in other contexts such as programming, where color is used for syntax highlighting



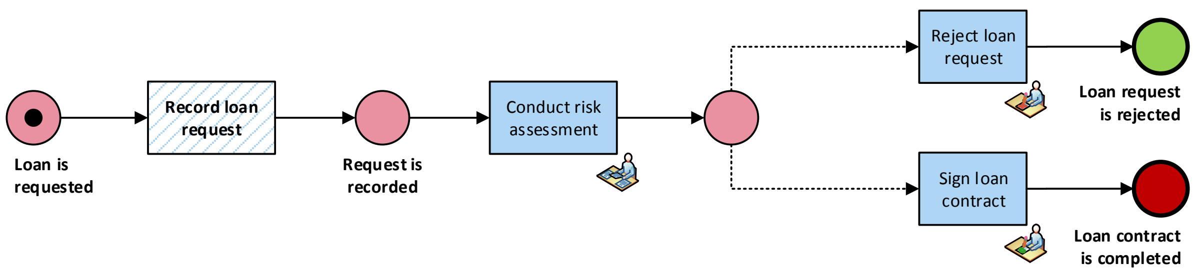
(a) Loan request as a Business Process Model and Notation (BPMN) diagram.



(b) Loan request as an Event-driven Process Chain (EPC).



(c) Loan request as a Petri net.



(d) Loan request as a Petri net with use of secondary notation.

Figure 2.6: Sample business process model. Source: based on [Men08, p. 9].

[Sar15]. An effective use of secondary notation is important—and somewhat of an art rather than just a mechanical procedure—as it allows clarifying the meaning of a visual representation without changing its underlying semantics [Moo09, KRM16]. The use of secondary notation is illustrated in Figure 2.6d, albeit in an arbitrary fashion not intended to improve the quality of the model. Here, pink and blue background colors are used for places and transitions respectively, the final places of the model are green and red depending on the outcome of the process and have a thicker border, additional icons are used to highlight the meanings of individual transitions, one activity has been given a random texture, and two arcs are drawn as dashed rather than solid lines.

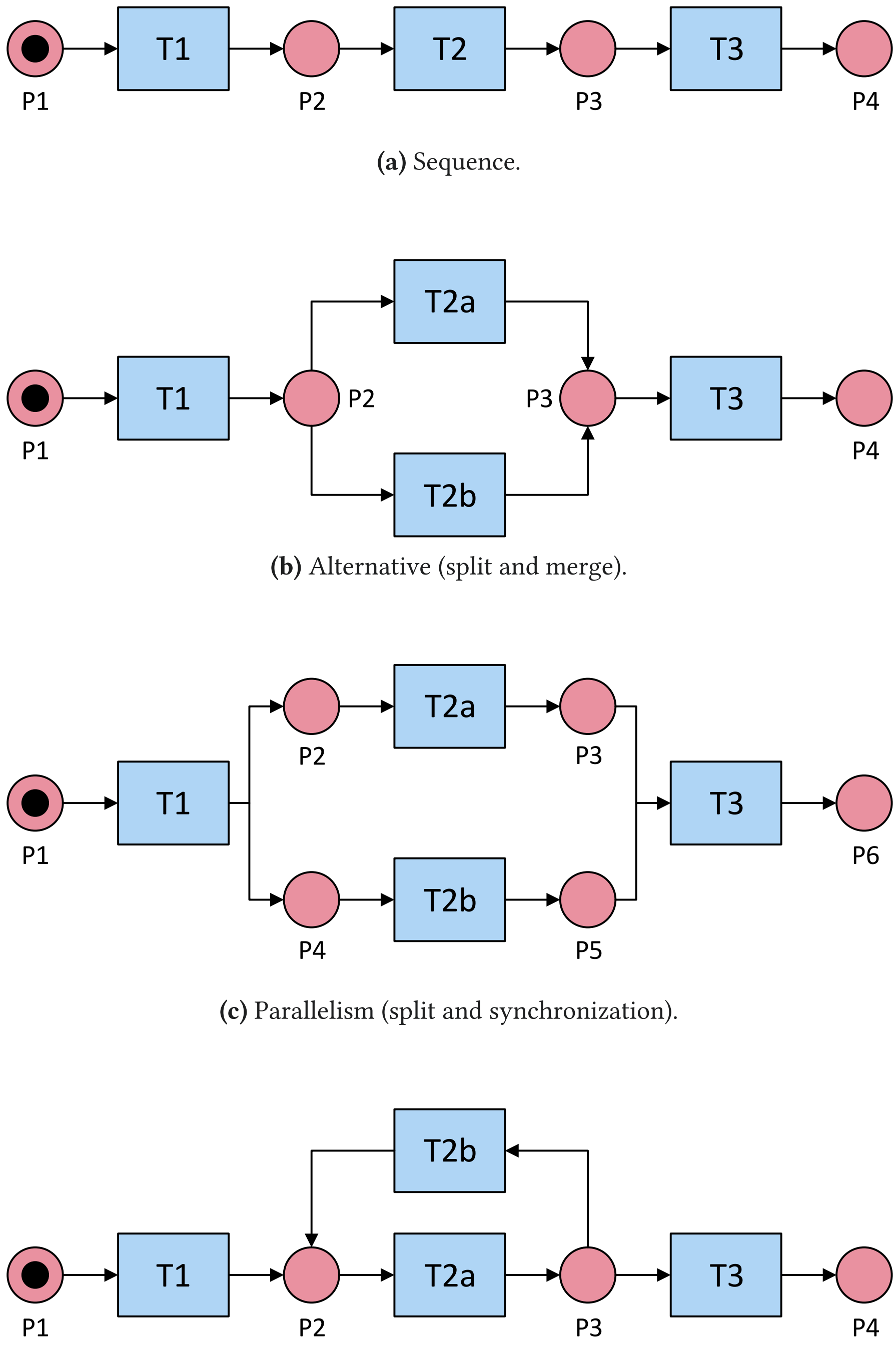
Control-flow. Looking at real-world processes, many complex interrelationships between the activities to be executed can be observed. In the most simple case, activities are executed in a *sequence* with fixed order. Taking the assembly of a car for example, the car body must be constructed before it can be painted, and the body must be painted before the car can be fitted with its interior equipment. In other cases, choices between execution *alternatives* must be considered. For instance, whether a navigation system is built into a car depends on whether this configuration option was chosen by the customer, and thus not all instances of the assembly process will include this activity. Furthermore, sometimes it may be possible to execute certain tasks in *parallel* until a point at which the process must be *synchronized*. For example, the body and the engine of a car can be built separately from each other until the point at which the latter is integrated into the former. Finally, some processes may require *iterations* that cause the repeated execution of previous activities. In the car assembly process, this may be the case if a manufacturing error occurs and corrective actions must be implemented. To specify such interrelationships, process activities must be connected by arcs in particular ways, thereby yielding the so-called *control flow* of the process [KtHvdA03].

The four types of control flow just described are illustrated as Petri nets in Figure 2.7. Due to the simplicity of a sequential flow, no further explanation is required for Figure 2.7a. In case of the alternative depicted in Figure 2.7b, after the execution of $T1$ has moved the token from $P1$ to $P2$, only $T2a$ *or* $T2b$ may fire, as the respective operation consumes the token required for that purpose. After one of the alternatives has been performed, the process is merged again in $P3$. As for the parallel flow modeled in Figure 2.7c, the execution of $T1$

places a token in both $P2$ and $P4$, and thus $T2a$ *and* $T2b$ become enabled and can be executed independently of each other. After being split up, the process is synchronized again by $T3$, which requires both parallel activities to have been finished, thus placing tokens in $P3$ and $P5$. Finally, the iteration shown in Figure 2.7d also incorporates an alternative choice, so that once a token has been placed in $P3$, the process may either continue via $T3$, or iterate back to $P2$ via the execution of $T2b$. Whereas different types of control flow are modeled in Petri nets implicitly through the manner in which arcs connect places and transitions, other modeling language define explicit notational symbols for deviations from sequential flow, which are typically called *connectors*, *gateways*, or *routing symbols*. For instance, gateways in BPMN are depicted as diamonds, whereas connectors in EPC are drawn as circles as shown in Figure 2.6. Typical types of connectors include *XOR* (exclusive OR) to model alternatives, *AND* to model parallelism, and *OR* (inclusive OR) to model alternatives where multiple activities can be executed as well. When using such elements, it is common to distinguish between *split* and *join* nodes which begin and terminate a block of activities with a certain behavior respectively. Further information on various types of control flow can be found in publications on *workflow patterns* [vdABtHK00, vdAtHKB03].

Modeling Method

As Becker et al. remark, business process modeling itself is a construction process [BPV12]. Hence, it requires the definition of a procedure that illustrates how modeling languages can be used effectively and efficiently [Men08]. Considering the structure of this process is important, as it has been shown that *how* modelers work on models has a significant impact on the quality of the outcome [CVR+12]. In relevant literature, various conceptualizations of the *process of process modeling* can be found. For instance, Soffer et al. propose that two distinct phases can be distinguished [SKW12]. In the first phase, the modeler creates a mental model of the domain of interest. In the second phase, this mental model is then mapped to constructs of the utilized modeling language, which yields a model artifact. Consequently, the syntax of a modeling language affects how modelers can reason about a domain and the shape of the mental model that they form. A similar model is presented by Pinggera et al., who refer to the two phases as *comprehension* and *modeling*, respectively [PZW+12]. However, the authors also add a third phase,



(a) Sequence.

(b) Alternative (split and merge).

(c) Parallelism (split and synchronization).

(d) Iteration.

Figure 2.7: Four types of process model control flow.

reconciliation, during which modelers modify the model to increase its understandability, for instance by renaming activities and utilizing secondary notation. This process is described as iterative, meaning that process models are successively refined. Finally, a more extensive process consisting of eight activities in four iterative phases has been proposed for general information modeling in [FvdW06] and adapted to BPM in [Men08]. Here, the authors distinguish between *elicitation*, *modeling*, *validation*, and *verification*. While such conceptualizations are helpful for reasoning about the process of process modeling on an abstract level, further research is required to better support modelers in enacting this process [SKW12]. Besides procedure models that focus on the creation of individual models, there also exist modeling methods that provide guidance for conducting entire process modeling projects that consider business processes within from multiple perspectives. One such approach is the Horus Method, which is subdivided into three phases that are described in Section 5.2.

Modeling Tool

Theoretically, there is no reason why business process modeling cannot be conducted using pen and paper. In practice, however, process modeling software tools such as the Horus Business Modeler (HBM) shown in Figure 2.8 are of crucial importance, since they provide many features that promise to reduce the costs and to improve the quality of modeling projects [IEH99]. Furthermore, empirical research suggests that tool functionality can increase the satisfaction of modelers with the utilized modeling language [Rec12b]. Such features include, for instance, a central model repository for redundancy-controlled model management, navigation capabilities that allow creating links between different models, the integration of additional modeling perspectives (e. g., data objects or organizational structures), providing additional metadata about process elements (e. g., processing times, costs, quality), modifying the set of available language constructs, and helping users to follow guidelines for maintaining model quality [BKR03, MRvdA10, Rec12b]. In the context of collaborative process modeling, tools may further provide support for awareness, communication, coordination, group decision making, and team and community building [MRW12, RMH13]. The importance of tool support is even greater when modeling business processes "in the large" (cf. [HFL$^+$11]), where they can help with evaluating, filtering, designing, and presenting large

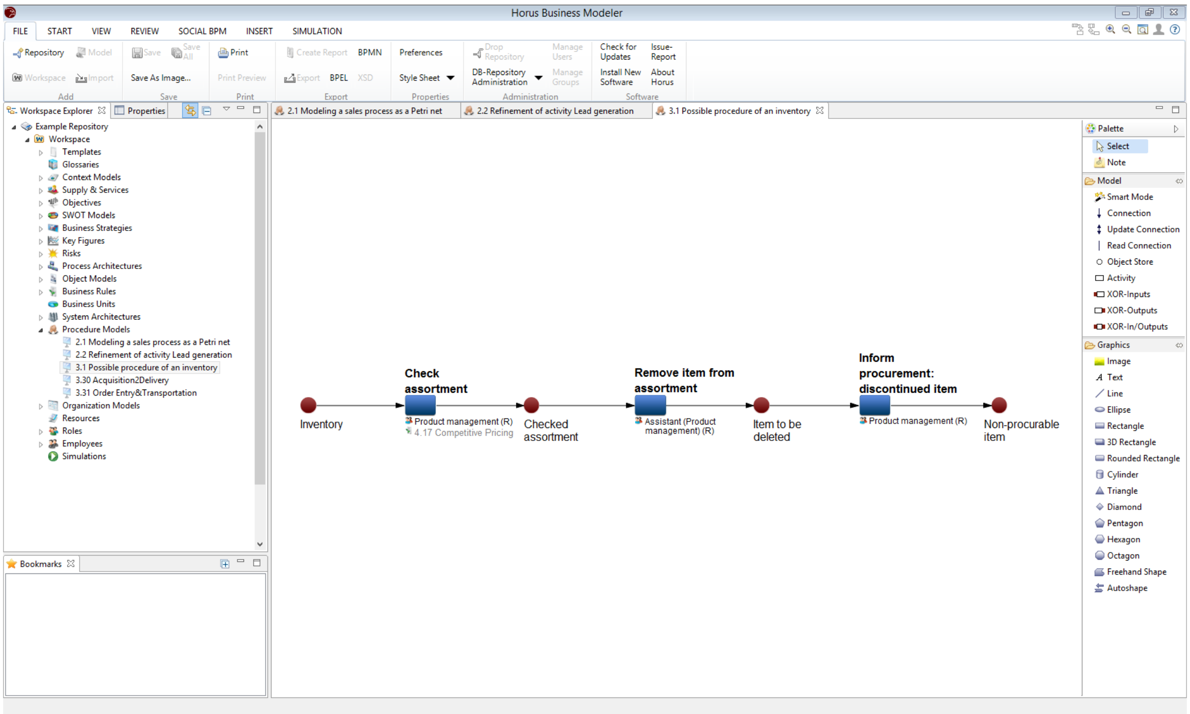


Figure 2.8: Sample business process modeling tool: Horus Business Modeler.

collections of models [LRvdA+11]. For additional information about various process modeling tools, the reader is referred to the survey presented in [AS07] as a starting point.

Modeling Context

As illustrated in Figure 2.5, business process modeling does not occur in a vacuum, but is embedded in a specific modeling context. Following Wand and Weber, three of the most critical contextual factors are individual difference factors, task factors, and social agenda factors [WW02a]. Firstly, *individual difference factors* refer to the experience, training, cognitive capabilities, and other skills of the involved process modelers. Generally speaking, two important roles can be distinguished: *domain experts* and *method experts* [RRvdW14]. Whereas the former have superior knowledge regarding the domain to be modeled but lack the required methodological expertise to utilize modeling methods and tools, the skill set of method experts is inverted [RFME11]. Secondly, *task factors* refer to the particular purpose for which business process modeling is conducted (i. e., organizational design or application system design as mentioned in the previous section) and the requirements that this imposes on the model artifacts. Lastly, *social agenda factors* refer to the wider context in which a process modeling project is embedded and the organizational change that it may cause.

2.4 Business Process Management

Today, businesses are faced with many difficult challenges, such as "a fast-moving business environment with changing customer needs and expectations, fast-evolving technologies and product life-cycles, strong globalization effects, accelerating innovation, and increasing digitization of products" [SB10, p. 239]. Therefore, enterprises must be willing to constantly revisit and improve their way of doing business and adapt their use of IS to support evolving business processes. This can be accomplished through Business Process Management (BPM), a holistic approach for maintaining business performance by managing business processes [Ham10] that "includes concepts, methods and techniques to support [process] design, administration, configuration, enactment, and analysis [...]" [Wes07, p. 5]. To that extent, BPM activities may be supported through the utilization of software systems called *business process management systems* [Wes07]. When carried out successfully, BPM enables businesses to "create high-performance processes, which operate with much lower costs, faster speeds, greater accuracy, reduced assets, and enhanced flexibility" [Ham10, p. 7]. The main purpose of BPM thus lies in the "continuous improvement of corporate strategies" [Neu09, p. 167] by transforming business processes through incremental or radical change [HC93, RvWML10].

In many (text)books, introductory articles, and surveys, Business Process Management is understood and presented as an iterative cycle—the so-called *BPM life cycle* of multiple interconnected activities (cf. [BKR03, Wes07, Ham10, MJ10, SVOK12, vdA13]). Most of these life cycles are conceptually very similar and build on the classic "Deming cycle" consisting of the steps *Plan*, *Do*, *Check*, and *Act* [Dem00]. Based on the integration and extension of the work of other authors, zur Muehlen makes one such proposal that is depicted in Figure 2.9 [zM04]. In this model, the following activities are distinguished [zM04, Men08]:

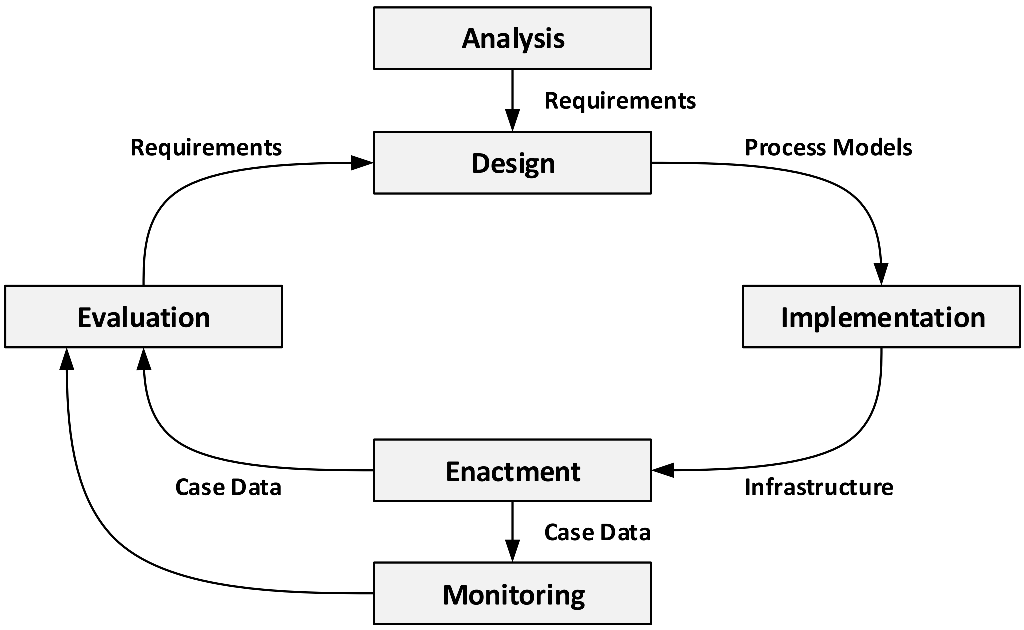


Figure 2.9: BPM life cycle. Source: based on [zM04, p. 86] as adapted by [Men08, p. 5].

- **Analysis**. The beginning of the cycle starts with an analysis of the organizational and technical environment in which BPM shall be conducted [zM04, Wes07]. This results in a collection of requirements (e. g., performance goals) for subsequent activities [Men08].

- **Design**. In the second step, business process models describing process activities, their order, and the resources and organizational roles on which they depend are specified [Men08]. The resulting models may be analyzed to ensure that they are syntactically correct, are free from formal errors such as deadlocks, and possess other desired properties [Wes07]. Furthermore, simulation can be employed to examine whether the designed processes meet predefined performance targets regarding, e. g., time, cost, and quality.

- **Implementation**. After the design phase, business processes are prepared for enactment. For processes whose execution is supported through an IS, this includes translating processes into a machine-readable format, preparing the technical infrastructure, and integrating the latter with any surrounding application systems [zM04]. Furthermore, affected employees must be trained to work with the new implementation [Men08].

- **Enactment**. Once the implementation has been finished, instances of business processes can finally be executed. This produces logs with case data about each process instance that form the basis for the final two activities.
- **Monitoring**. Simultaneously to enactment, each individual process instance is monitored and performance figures such as the lengths of waiting times and the utilization of resources are recorded [zM04]. If monitoring detects the existence of a problematic or otherwise exceptional situation, monitoring might result in the execution of appropriate counteractions [Wes07, Men08].
- **Evaluation**. In the final phase, an ex-post analysis is conducted by comparing aggregated performance figures with the original requirements [Men08]. Based on the results, potential for process improvement can be identified and incorporated into the next iteration of the life cycle.

Besides this procedural perspective, BPM can also be considered on a broader scale by accounting for its holistic nature as an approach addressing all aspects of an enterprise to manage its business processes [SVOK12]. From this perspective, BPM represents an organizational capability rather than just the adherence to a set of tasks organized as an iterative life cycle [RvB10]. Based on an extensive literature review on BPM maturity models and subsequent Delphi studies, Rosemann and vom Brocke conclude that holistic BPM consists of the following six core elements [RvB10].

- **Strategic alignment** refers to the alignment of BPM efforts with the overarching strategy of an organization to enable a continuous improvement of business process performance. This includes the creation of a strategy-driven process improvement plan, conducting a bidirectional analysis between strategy and business processes, establishing a high-level enterprise process architecture, defining metrics and key performance indicators for process outputs, and considering the priorities of relevant internal and external stakeholders.

- **Governance** is related to accountability in the context of BPM. This includes the creation of decision-making processes that clarify when, how, and by whom decisions related to BPM can be made, the description of roles and responsibilities, performing measurements for the metrics defined by strategic alignment, documenting process management standards and guidelines, and specifying remuneration and reward schemes.

- **Methods** refer to the tools and techniques that are employed across the various activities of the BPM life cycle. This includes methods for process design and modeling, process implementation and execution, process control and measurement, process improvement and innovation, and project management.

- **Information technology** denotes all types of hardware and software that support and enable BPM activities by complementing the aforementioned methods.

- **People** includes all human resources who contribute to improving business performance by continuously applying and improving their skills. This core element addresses the skills and expertise that are required for process enactment and process management, measures for learning and education as well as collaboration and communication.

- **Culture** refers to the "collective values and beliefs that shape process-related attitudes and behavior to improve business performance" [RvB10, p. 119]. This core element is concerned with the responsiveness of an organization to process change, about the values and beliefs of individuals regarding processes, the attitudes and behaviors of individuals regarding BPM, the attention of leadership to BPM, and organizational communities of BPM practice.

Each of these six core elements represents a critical success factor that can impact the success of BPM in various meaningful ways. Many other authors have addressed this topic as well, and thus BPM literature offers a plethora of publications that specifically focus on how the success of BPM can be ensured and which factors contribute to the former [AF08, RB10, Trk10, PRR13, ST13, vBSR+14]. Due to the particular focus of this doctoral thesis on business process *modeling*, Business Process Management as a whole will not be discussed any further in the following. For additional information on the topic, the reader is thus referred to any of the numerous reviews [KLL09, Neu09, vdA12a, vdA13] and textbooks [BKR03, Wes07, DLMR13].

2.5 Business Process Graphs

As previously discussed, there are many different approaches for creating models of business processes (cf. the representation types in Figure 2.3), and even focusing on diagrams as the most common, formalized means of representation, a wide variety of different methods and standards exists (cf. the selection of modeling languages presented in Section 2.3). This leads to the problem that a discussion of many model-specific properties—including the quality of business process models, which is a core concern of this thesis—can either only be valid for a single modeling language, or requires considerable redundancies to address all major modeling approaches. However, looking at sample Petri nets, BPMN diagrams, and EPCs side by side, it can easily be seen that many languages are conceptually similar in the sense that they depict business processes as sets of nodes of varying types connected by arcs [PBD+13]. Therefore, the observation is leveraged that many business process models can be described as general graphs consisting of nodes, edges, and labels [DDvD+11]. This is a common practice in research on conceptual models in general and business process models in particular, and is also done, e. g., in works on model pattern matching [PBD+13, DSDB15], model similarity analysis [DDvD+11], linguistic model analysis [Leo13], model verification and error prediction [Men08], and process model repositories [LRvdA+11].

A language-independent model representation based on graphs is also called a *canonical form* and provides further advantages such as standardization, efficiency, interchangeability, reusability, and flexibility [LRvdA+11]. However, the main purpose of the canonical form presented here is to allow for the definition of quality metrics in Section 2.7 independently of the underlying process modeling language. Following [DDvD+11], the canonical form of a business process model will henceforth be referred to as a *business process graph*. Operationalizing the canonical form requires *transformation* procedures that take models of arbitrary process modeling languages as an input and

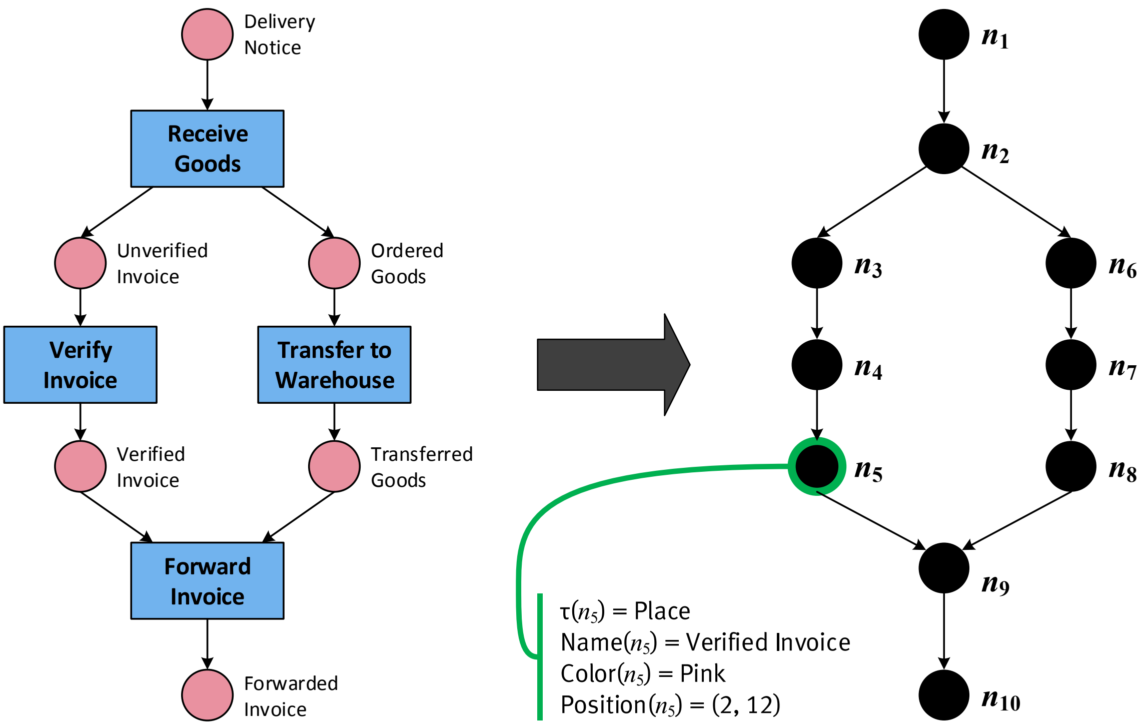


Figure 2.10: Transformation of a sample Petri net into a business process graph.

convert them into business process graphs. As the specifics of how such a transformation can be carried out are outside the scope of this thesis, they are not presented here. Nevertheless, it should be noted that some of the more specific intricacies of certain modeling languages might be difficult (or impossible) to transform, and thus one of the underlying ideas of using a canonical form is to omit such constructs [LRvdA$^+$11]. An example depicting a business process model (left) and its corresponding representation as a business process graph (right) can be seen in Figure 2.10. The following paragraphs provide a formal specification for business process graphs that will be used to unambiguously define quality metrics in Section 2.7. Where appropriate, these formalisms will be substantiated by example of the model and corresponding graph in Figure 2.10

A business process graph G is defined as a tuple $G = (N, A, T, \tau, \Lambda)$[1], where

[1] In general literature on graph theory, e. g. [Die10], graphs are typically defined as tuples $G = (V, E)$ so that the set of vertices V corresponds to N and the set of edges E to A. However, this thesis adheres to the practice often employed in BPM literature to use more domain-specific set identifiers.

- N is a non-empty set of *nodes*.
- $A \subseteq N \times N$ is the set of unweighted, directed *arcs* representing process control flow.
- $E = N \cup A$ shall serve as a shorthand for the union of nodes and arcs, representing all model *elements*.
- T is a set of possible *types* of the elements $e \in E$. Sample types include *Activity*, *Event*, *Place*, *Transition*, and *Arc*.
- $\tau : E \rightarrow \mathcal{P}(T) \setminus \varnothing$ is a function mapping each element to a non-empty subset of types. Consequently, an element may be assigned multiple types at the same time, e. g., *Activity*, *Transition* and *XorSplit*.
- $E_t = \{e \in E \mid t \in \tau(e)\}$ refers to the set of all elements in the graph of type t (e. g., the set of all activities).
- $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_l\}$ represents the set of all *labeling functions* through which the elements of the graph are assigned additional properties carried over from the source process model by the transformation procedure.

Example. Looking at the sample Petri net and graph depicted in Figure 2.10, the set of nodes is $N = \{n_1, n_2, \ldots, n_{10}\}$, the set of arcs is specified as $A = \{(n_1, n_2), (n_2, n_3), (n_2, n_6), \ldots, (n_9, n_{10})\}$, and the set of types contains (at least) the elements $T = \{Place, Transition, Arc, AndSplit, AndJoin\}$. Examples for the assignment of particular types to model elements include $\tau(n_5) = \{Place\}$ and $\tau(n_9) = \{Transition, AndJoin\}$. As an example for E_t, the set of all places can be specified as $E_{Place} = \{n_1, n_3, n_5, n_6, n_8, n_{10}\}$. Lastly, the set of labels in Figure 2.10 is $\Lambda = \{Name, Color, Position\}$ and their concrete assignment for the single node n_5 is $Name(n_5) =$ *Verified Invoice*, $Color(n_5) =$ *Pink*, and $Position(n_5) = (2, 12)$, with the latter denoting its location in two-dimensional space.

Labels. To carry over business properties (e. g., regarding time, cost, or quality), language-specific attributes, and other metadata from their original source models to business process graphs, labeling functions and labels are used. In general terms, the former assigns values of one particular type to the elements of a model, and the latter represents the concrete value the function assumes for a specific element. More formally, every labeling function $\lambda_i : E \rightarrow L_i, 1 \leq i \leq l$ is a *partial* function mapping elements to a set of possible labels L_i. The fact that labeling functions are partial means that not all elements are necessarily assigned a label by all labeling functions. In particular, some labels may only be valid or relevant for particular types of elements (e. g., all elements may have a name, but processing times are only relevant for activities) so that for every type t there is a subset Λ_t of the labeling functions in Λ valid for that type. On this basis, for any given labeling function $\lambda \in \Lambda$, the following special subsets of elements can be defined:

- Set of all elements that are labeled: $E_\lambda^{act} = \{e \in E \mid e \in dom(\lambda)\}$
- Set of all elements that can be labeled: $E_\lambda^{pot} = \{e \in E \mid \lambda \in \Lambda_{\tau(e)}\}$
- Set of all elements that can be labeled but are not: $E_\lambda^{mis} = E_\lambda^{pot} \setminus E_\lambda^{act}$

In the example shown in Figure 2.10, at least three labeling functions can be identified: $Name$, $Color$, and $Position$. Assuming that arcs do not have a $Position$ and their visualization is determined by the locations of their source and target nodes, the set of labels for the type Arc is limited to $\Lambda_{Arc} = \{Name, Color\}$. Thus, $\Lambda_{Place} = \Lambda_{Transition} = \{Name, Color, Position\}$. As all places and transitions are assigned a name, but none of the arcs are, $E_{Name}^{pot} = E$, $E_{Name}^{act} = E_{Place} \cup E_{Transition}$ and $E_{Name}^{mis} = E_{Arc}$.

Connectivity. For a given node $n \in N$, the set of its *incoming arcs* and *outgoing arcs* can be respectively defined as

$$n_{in} = \{(x, n) \mid x \in N \wedge (x, n) \in A\} \quad (2.3)$$

$$n_{out} = \{(n, x) \mid x \in N \wedge (n, x) \in A\} \quad (2.4)$$

This allows for the specification of three quantitative measures for each node, namely the number of incoming arcs (*in degree*) as $deg_{in}(n) = |n_{in}|$, the number of outgoing arcs (*out degree*) as $deg_{out}(n) = |n_{out}|$, and the sum of the two (*degree*) as $deg(n) = deg_{in}(n) + deg_{out}(n)$. On this basis, two specific subsets of nodes can be distinguished: the start nodes (*sources*) of a process which have no further preconditions, and its end nodes (*sinks*) that represent the final process outcome. Both of these sets can be respectively defined as

$$N_{start} = \{n \in N \mid deg_{in}(n) = 0\} \tag{2.5}$$

$$N_{end} = \{n \in N \mid deg_{out}(n) = 0\} \tag{2.6}$$

In the example presented in Figure 2.10, the incoming and outgoing arcs of node n_2 are $n_{in}(n_2) = \{n_1\}$ and $n_{out}(n_2) = \{n_3, n_6\}$, and correspondingly, n_2 has an in degree of 1, an out degree of 2, and an overall degree of 3. Furthermore, the sources and sinks of the process model depicted in the figure are $N_{start} = \{n_1\}$ and $N_{end} = \{n_{10}\}$.

Paths. A *path* between two nodes $n, m \in N$ written as $n \rightsquigarrow m$ is said to exist if there is at least one sequence of nodes $x_1, \ldots, x_k$ so that $n = x_1$ and $m = x_k$, and for all $i \in 1, \ldots, k-1$ there is an arc $(x_i, x_{i+1}) \in A$. As G is unweighted, the *length* of a particular path is $k-1$, and out of the set of possible paths between two nodes $P(n, m)$, the one with the smallest length is also referred to as the *shortest path*. The length of the shortest path is also called the *distance* between two nodes and can be indicated as $d(n, m)$. Furthermore, the *diameter* of G is defined as the length of the longest path in the set of all shortest paths between any of its nodes, i. e.,

$$diam(G) = \max_{n,m \in N} d(n, m) \tag{2.7}$$

In the following, it is assumed that G is *weakly connected*, meaning that in the undirected graph G' that can be obtained from G by replacing all directed arcs with undirected arcs, there is at least one path $n \rightsquigarrow m$ for all pairs $n, m \in N', n \neq m$.

In Figure 2.10, two paths exist from the source of the business process to its sink, namely $n_1, n_2, n_3, n_4, n_5, n_9, n_{10}$ and $n_1, n_2, n_6, n_7, n_8, n_9, n_{10}$. As the length of both paths is 6, both possibilities are also a shortest path between the two nodes, and 6 is also the diameter of G.

Connectors. For business process graphs, no assumption is made about whether the source modeling language represents connector nodes explicitly (BPMN, EPC) or implicitly (Petri nets). This is possible due to the fact that the mapping function τ can assign more than one type to any node, so that it can, for instance, be typed as an *Activity* and an *XorSplit* at the same time, should its behavior match both. Based on the mathematical constructs defined so far, connector nodes are integrated into the canonical form as follows:

- $Conn$ is a set of *connector archetypes.*
- $T \supseteq T_{Conn} = \bigcup_{c \in Conn} \{t_{c_{split}} \cup t_{c_{join}}\}$ is a subset of *connector types* so that for each connector archetype, T_{Conn} contains both a *split connector type* $t_{c_{split}}$ and a corresponding *join connector type* $t_{c_{join}}$.
- For most common business process modeling languages, the set of connector archetypes is $Conn = \{and, xor, or\}$ and consequently $T_{Conn} = \{and_{split}, and_{join}, xor_{split}, xor_{join}, or_{split}, or_{join}\}$.
- $C = \bigcup_{t \in T_{Conn}} E_t$ is the set of *connector nodes.*
- For a given connector archetype $c \in Conn$, $S_c = E_{c_{split}}$ is the set of respective *split nodes*, $J_c = E_{c_{join}}$ the set of respective *join nodes*, and $C_c = S_c \cup J_c$ the union of both.

In Figure 2.10, only and connectors are used, so that one connector archetype suffices, i. e., $Conn = \{and\}$ and $T_{Conn} = \{t_{and_{split}} \cup t_{and_{join}}\}$. Furthermore, only two nodes cause a branching of the process control flow, namely n_2 with a parallel split and n_9 with the corresponding synchronization. Thus, the set of and split nodes is $S_{and} = \{n_2\}$, the set of and join nodes is $J_{and} = \{n_9\}$, and the sets of all and connector nodes and all connector nodes in total are identical, i. e., $C = C_{and} = \{n_2, n_9\}$.

2.6 Business Process Model Quality

The term “quality” is often used in everyday life to make valuating statements about products and services, such as books, clothing, movies, or games. It is associated with expressing opinions about whether something is “good” or “bad”. More formally, the quality of a thing can be defined as its “standard or nature [...] as measured against other things of a similar kind” [27]. Thus, quality is associated with the degree of excellence a certain object possesses and its superiority over other objects with lower quality [27]. Consequently, measuring the quality of things makes it possible to select the best representative from a set of alternatives with similar nature. Therefore, quality measurement is an important consideration in many different areas and there exists a plethora of standards, frameworks, and approaches that are geared towards managing and improving quality, such as the ISO 9000 family of standards [ISO15a, ISO15b], the ISO/TS 16949 standard for the automotive industry [IAT16], Total Quality Management [Cro79, Pow95], and Six Sigma [HS06].

Depending on the purpose for which a thing is used, the assessment of its quality may change, and thus something that is superior for a certain purpose may be inferior for another. Other definitions consider this property of quality explicitly, for instance by describing it as "conformance to requirements" [Cro79], the "degree to which a set of inherent characteristics fulfills requirements" [ISO15b], or simply "fitness for use" [JG99]. This can easily be illustrated by example of the purchase of a new car. Here, it is assumed that the prospective buyer can choose between two options: car A with a maximum speed of 100 km/h and an average fuel consumption of 5 liters/100km and car B with a maximum speed of 200 km/h and an average fuel consumption of 10 liters/100km. Considering the subjective requirements defined by the buyer, each of these two options may have the higher quality: option A for maximum fuel efficiency and option B for maximum speed. The assessment of both cars may change yet again depending on the importance of further criteria such as buying price, operating costs, or stowage space, thus further underlining the pragmatic nature of "quality".

Similarly, the quality of a business process model can be understood as its fitness for use [BRvU00] as well, i. e., its ability to satisfy certain requirements that depend on the purpose for which it will eventually be used (cf. Section 2.3). Thus, following Becker et al., a model is not just simply right or wrong, but either more or less appropriate for its intended purpose, and thus in the eyes of its user either of higher or lower quality [BPV12]. For instance, while being syntactically correct might be essential for a model intended to be algorithmically analyzed by a computer program, this is far less important for a model that is the outcome of a brainstorming session for process innovation. Ensuring a high quality of business process models is an important concern, as the quality of any decisions made on the basis of any model can be expected to also increase with a higher model quality (cf. [Moo05]).

Of course, the discussion of quality in the context of BPM is not just limited to models, but also addresses many other aspects of business processes [MRR09]. These issues will not be addressed in the remainder of this thesis, but are mentioned here in the interest of completeness. For additional information, the reader is referred to the mentioned sources as a starting point. Firstly, as indicated by Figure 2.9, an integral part of managing business processes lies in monitoring them after deployment so that their performance can be evaluated against predefined goals regarding, for instance, cost, time, and quality. Thus, the notion of quality can also be extended to business processes as a whole and individual process instances as they are being enacted. Relevant work in this area mostly addresses issues of business process measurement and improvement, e. g., [Nis98], [RL05], [HL14], [Loh15], and [Kro16, ch. 2.3.1]. Secondly, the quality of business process modeling languages clearly plays an important role for modeling projects, as the use of a "better" language can intuitively be assumed to also result in better models. A considerable portion of research in this area examines the quality of the primary notation of modeling languages [FMS09, GHA10, FD11, Fig12, FMS13], in particular with regards to the design of symbols for routing control flow [FMSR10, Rec12a, FRM13]. This is often done against the set of principles for effective visual notations defined by Moody, which include semiotic clarity, perceptual discriminability, visual expressiveness, dual coding, and graphic economy [Moo09]. Other authors have conducted representational analyses of modeling languages by analyzing which constructs and relationships they can express [LK06, RRIG09]. Lastly, researchers may also study the process of process modeling and its impacts on the quality of the resulting models, as for instance done by Claes et al. [CVR+12]. Another example for work in this area is "Quality of Modelling (QoMo)" [vBHPV07], a quality framework for process modeling connecting modeling goals to SEQUAL, a quality framework for business process models briefly discussed in Section 2.6.2. Further topics of interest concerned with quality in BPM include the quality of the employed modeling tools and methods, the involved modeling stakeholders, and the management support for modeling projects [MRR09].

The main purpose of this section lies in giving an overview over the most important aspects of business process model quality that are discussed in relevant academic literature. To that extent, Section 2.6.1 first addresses bottom-up quality metrics that can be used to measure individual quality aspects of a model. Afterwards, Section 2.6.2 provides information about top-down quality frameworks that provide an organized, systematic view on model quality. Lastly, quality guidelines that give human modelers actionable recommendations on how to create and manipulate models to maintain a high quality level are discussed in Section 2.6.3.

2.6.1 Quality Metrics

The literature offers a considerable amount of anecdotal evidence for the importance of measurement. For instance, the following quote is attributed to Galileo Galilei [Men08]: "Count what is countable, measure what is measurable, and what is not measurable, make measurable." Furthermore, DeMarco expresses his position that "[one] can't control what [one] can't measure" [DeM82, p. 3], and that inversely, "[rational], competent men and women can work effectively to maximize any single observed indication of success" [DeM82, p. 58]. According to Mendling, measurement in business process modeling can serve at least the purposes of understanding, control, and improvement [Men08]. However, business process model quality is still a rather new research area that has started to gain traction since 2006, and thus there is not yet a clear consensus about *what* it actually is that should be measured. Nevertheless, it is obvious that the quality of business process models is a multi-dimensional construct that is affected by various concerns such as completeness, correctness, readability, and complexity. Thus, it cannot be captured by a single metric alone [Men08]. Instead, quality metrics can be seen as the basic building blocks that allow researchers and practitioners to gain a more holistic understanding of the attributes that "good" models should possess.

In the following, a distinction between the terms *quality metric* and *measurement procedure* will be made. Whereas the former refers to a particular quality aspect of business process models on a conceptual level, the latter specifies how a metric is operationalized, i. e., measured. This distinction is important, as some metrics may be measured in different ways. For instance, the "size" of a process model may either be computed as the number of activities it contains, or as its total number of elements regardless of their type [CMNR06]. Adopting the terminology used in [OBS12], two types of measurement procedures will be distinguished: *absolute* procedures that measure the full extent of a metric, and *relative* procedures that put the former in relation to an extremal reference value, e. g., the size of the model. Formally, the set of absolute measurement procedures is defined as $A = \{a_1, a_2, \dots, a_n\}$ with $a_i : G \rightarrow \mathbb{Q}, 1 \leq i \leq n$, i. e., A is a set of functions that map a business process graph G to a rational number. Building on this, we analogously define $R = \{s_1 \circ a_1, s_2 \circ a_2, \dots, s_n \circ a_n\}$ with $s_i \circ a_i : \mathbb{Q} \rightarrow [0, 1], 1 \leq i \leq n$ as the set of relative measurement procedures with compositions of functions $(s_i \circ a_i)(G) = s_i(a_i(G))$, where s_i "makes" a_i relative, i. e., standardizes it to the interval $[0, 1]$. The shorthand r_i is henceforth used synonymously for $s_i \circ a_i$. Furthermore, this standardization ensures that relative measurements yield 0 as the worst value of a quality metric and 1 as its best value. An example for an absolute measurement procedure is a function counting the number of edge crossings in a drawing of a given G, whereas the corresponding relative measurement function would yield 1 if there are no edge crossings at all and 0 if all possible edge crossings actually exist. This is illustrated by the model depicted in Figure 2.11, which exhibits 2 out of 33 possible edge crossings, the latter number being calculated using a heuristic specified in [Pur02b]. Consequently, the absolute measurement for the respective graph yields 2, whereas the relative measurement is $1 - (2/33) = 0.94$.

The discussion of concrete quality metrics for process models can be traced at least as far back as the 1990s [LY92, KM96, Mor99]. However, it was not until 2006 that the volume of research conducted in this area experienced a dramatic increase, starting with studies such as [Car06], [CMNR06], and [GL06]. Generally speaking, quality metrics for business process models can come from a variety of sources, such as software complexity [LY92, CMNR06] and graph drawing aesthetics [Pur02b, EJS10]. Additionally, some authors have designed original metrics specifically for business process models [Car06, Men08, VRM$^+$08]. Finally, quality metrics can be defined on at least three different levels of specificity (cf. [BRvU00]). Firstly, some metrics may be valid for all types of conceptual models, including process models, data models and organizational diagrams. Secondly, a subset of metrics may only apply to process models, but without any restrictions regarding the underlying modeling language. Lastly, metrics may also be specific to a particular modeling language, if they are based on particular characteristics of the former that other

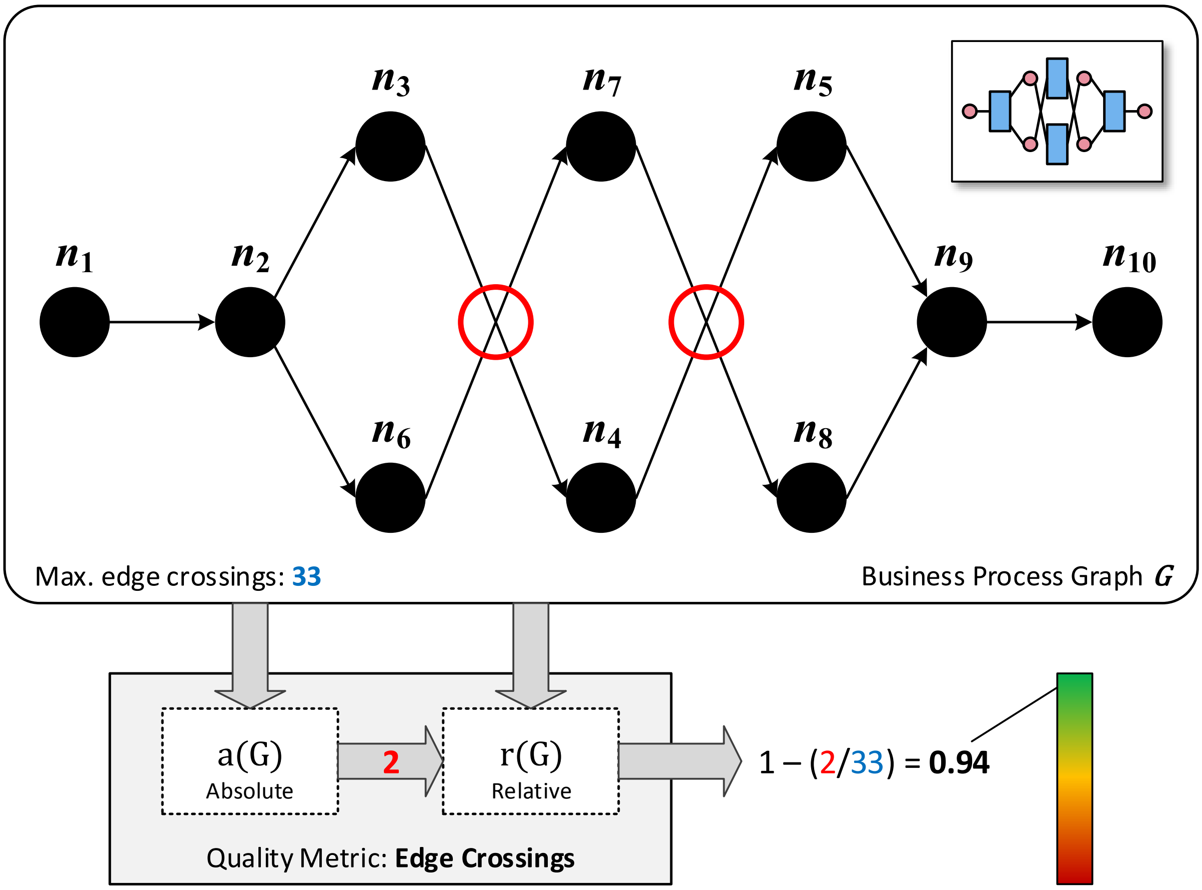


Figure 2.11: Quality metrics example: Edge crossings.

languages do not possess. Concrete quality metrics play an important role for the application part of this doctoral thesis, and thus a detailed discussion of various metrics is presented in Section 2.7.

2.6.2 Quality Frameworks

The term "framework" is often used, but seldom defined. For instance, Moody's seminal publication on quality frameworks for conceptual models contains the term 116 times (excluding references), but assumes that its meaning is inherently clear to the reader. Looking at the Oxford English dictionary, a framework can generally be understood as "an essential or underlying structure" or "a conceptual scheme or system" [28]. Based on this definition, the term "business process model quality framework" will further be understood as any conceptual scheme aiming to give structure to the notion of the quality of business process models. Based on the ISO/IEC 9126 standard, Moody proposes the following *structural requirements* for such a framework [Moo05, p. 256]:

1. Conceptual model quality should be decomposed into a hierarchy of quality characteristics, subcharacteristics and metrics.
2. Single-word labels should be used for each quality characteristic and subcharacteristic, using commonly-understood terms.
3. Each quality characteristic and subcharacteristic should be defined using a single, concise sentence.
4. Metrics should be defined for measuring each subcharacteristic.
5. Detailed procedures should be defined for conducting quality evaluations.

From these requirements, a generic framework structure as depicted in Figure 2.12 can be derived. On the lowest level of this framework, individual quality metrics together with their respective measurement procedures reside. For a given input model, each of these metrics delivers a value between 0 and 1 as described in the previous section. On the next higher level, these elementary building blocks are summarized to form so-called *quality subcharacteristics*.

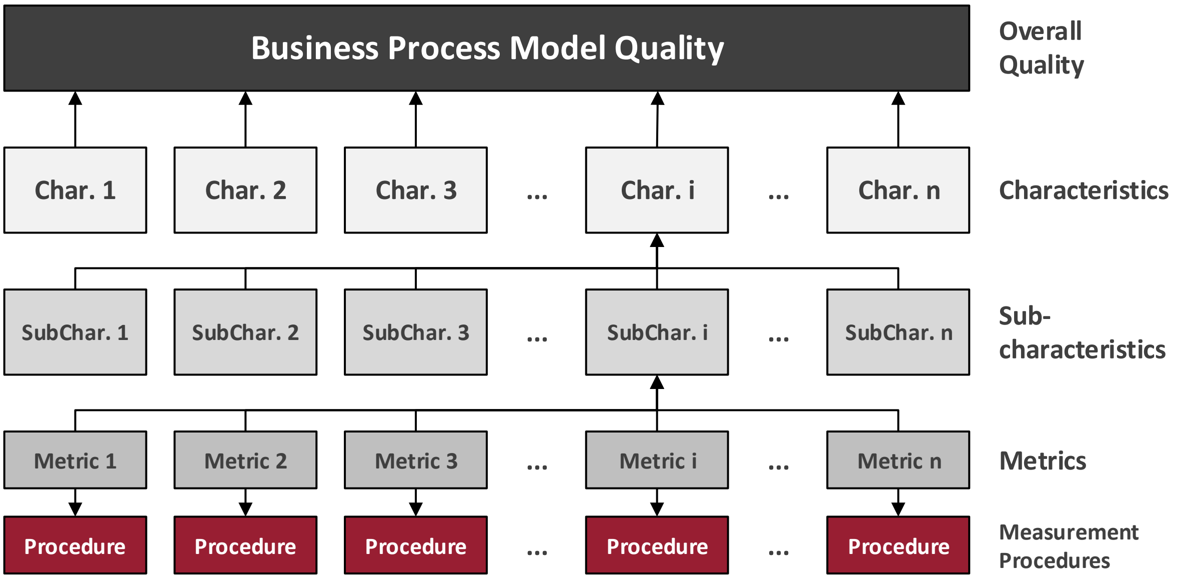


Figure 2.12: Structure of model quality frameworks. Source: based on [Moo05, p. 254].

Sample subcharacteristics of the Quality Marks, Metrics, and Measurement (3QM) framework include correctness, relevance, completeness, text syntax, and understandability [OBS12]. One further level above, subcharacteristics are grouped together into *quality characteristics*, which represent higher-level quality concerns such as syntactic, semantic, and pragmatic quality [LSS94, RMR10, OBS12]. Finally, at the highest level of the framework stands the *overall quality* of entire business process models. As concrete measurement only occurs at the lowest level of this generic framework, weighting functions are required between all levels so that quantitative quality indicators can also be determined at the higher levels.

Besides these structural demands, further *operational requirements* (i. e., requirements regarding the operational use of a quality framework) can be specified. In particular, Becker et al. point out that a shortcoming of many current approaches is their presumption that the quality of a model is independent of the purpose for which it is used [BKR03]. However, it is clear that this is not the case. For instance, whereas a model created for organizational documentation may have specific needs regarding readability and understandability, a technical model created for simulation may instead require the complete specification of simulation-specific information such as processing times and costs. Thus, a model that can be considered "perfect" for one purpose is not necessarily well-suited for another. Similarly, the perceived quality of a model may also vary depending on its appropriateness for the audience for which it was created [BKR03]. For example, based on [Pet06], Reijers and Mendling argue that the negative effects of a poorly laid-out process diagram affect modeling novices more than experts [RM11]. One simple way to address these two requirements would be to weigh metrics and characteristics at the various levels of the frameworks in dependence of the modeling purpose and target audience, although coming up with appropriate weights is a difficult task [OBS12]. Finally, a quality framework can only be considered useful if it actually enables modelers with different backgrounds to create high-quality models. However, this cannot be said without doubt for most of the frameworks introduced below, which often remain too abstract to be operationalized with ease [MRvdA10], although some authors claim that their frameworks have been designed with usability in mind [RMR10, OBS12].

The next paragraphs will give a brief overview of the quality frameworks that are frequently discussed in academic BPM literature. For a more comprehensive list of frameworks addressing also other types of conceptual models, the reader is referred to [Moo05] as a starting point.

Semiotic Quality Model (SEQUAL)

One of the most comprehensive and influential frameworks used in the BPM context is the Semiotic Quality Model (SEQUAL) framework, which was first proposed by Lindland et al. [LSS94, KLS95] and later refined by Krogstie et al. [KJ03, KSJ06]. While SEQUAL is conceptualized for the evaluation of all types of conceptual models, specializations of the framework have also been applied to various domains [Kro12], including business process modeling [Kro16]. Based on the observation that the process of modeling is essentially equivalent to using some language to make statements about things [LSS94], it is linked to concepts from linguistics and semiotics [KSJ06], and this understanding has also influenced other quality frameworks [RMR10, OBS12]. Besides the models themselves, the SEQUAL framework also considers other aspects that influence quality assessment, including the actors that work with the models, their knowledge, and their model interpretations, the modeling language, the domain expressed in the model, and the modeling goal [KSJ06]. Based on these entity sets, the following quality types are distinguished [KJ03, Kro16]:

- **Physical quality** is concerned with the representation of process models as physical artifacts and has two goals, *externalization* and *internalizability*. Externalization means that modelers must be enabled to represent their knowledge as a formal process model. In contrast, internalizability denotes the availability of process models as artifacts that can be interpreted by users.

- **Empirical quality** is related to the *readability* and layout of the model and addresses the question whether it can be easily read. This is just a property of the model itself, and as such not related to the actor who is using the model. In particular, empirical quality is concerned with the *secondary notation* of a model, i. e., the use of visual variables such as position, size, color, and texture [WGK10], to influence its readability without altering its meaning [Pet95, Pet06]. Furthermore, as many types of models—especially process models—are often visualized as graph drawings consisting of nodes and edges, empirical quality also is strongly connected to the field of graph drawing aesthetics (see., e. g., [Pur02b]).

- **Syntactic quality** denotes the correspondence of a process model with the syntactical rules imposed by a specific modeling language, i. e., its *syntactic correctness*. It is of special importance for processes whose enactment should occur with (partial) IT-supported automation and can be assured by software providing means for the detection, prevention, and recovery of errors. However, Krogstie et al. point out that the creation of syntactically incomplete models should still be possible to support learning and enable discussions [KSJ06].

- **Semantic quality** is given if a model exhibits *validity* and *completeness* with respect to its purpose and the domain that it reflects. A model is valid if it contains only relevant and correct statements about the domain. In turn, a model is complete if it contains *all* statements that are relevant and correct about the domain. In practice, perfect semantic quality may be difficult to achieve and must thus often be judged in relation to the modeling goal, which leads to the notion of *feasibility*, i. e., a relaxation of validity and completeness when further modeling is not considered beneficial anymore [KSJ06].

- **Pragmatic quality** is related to *comprehension* (not to be confused with mere comprehensi*bility*) and is given if a process model is consistent with its interpretation through human individuals, thereby allowing the reader to learn from it. This is influenced by two types of factors: personal factors such as modeling experience and education, and model characteristics [MRC07]. Research indicates that the former has the highest influence on understanding, and thus the importance of educating and training modeling participants must be stressed [RM11]. Focusing on the latter, negative effects on comprehension have been observed, e.g., for model size, average connector degree and model density [MRC07]. Measures that can improve pragmatic quality include model transformation, simulation, animation, and explanation generation [KSJ06].
- **Social quality** is concerned with *feasible agreement*, which requires participants of the modeling effort to reach a shared understanding about a model as the result of social learning. It is thus characterized by a convergence of the knowledge and model interpretations of all actors. This does not necessarily imply that a consensus is reached, but merely that inconsistent views are resolved where the benefits of resolution exceed the costs. Ensuring social quality not only reduces inconsistencies, but can also help with the detection of harmful contributions made by malicious actors. This is of special importance when developing information systems, as disagreement about models can lead to significant technical work that is hard to redo. Measures for achieving high levels of social quality include model integration, conflict resolution, and using argumentation support systems.
- **Deontic Quality** concerns the fact that models are typically only a means to an end, and thus relates to the question whether they contribute to pre-defined modeling goals. Feasibility plays an important role in this context, as in most realistic scenarios it will not be possible to create a "perfect" model, but rather only one that is "good enough" for its purpose. As such, deontic quality is also related to the notion of "fitness for use" [BRvU00] that was previously mentioned.

Being a high-level conceptual framework, no concrete quality metrics are discussed in most publications about SEQUAL. However, in [Kro12] and [Kro16], Krogstie aims to fill this gap by connecting relevant literature to the quality types of the framework, but does not present corresponding measurement procedures in most cases, thereby preventing its operationalization without additional effort. In an empirical study examining the applicability of SEQUAL to process modeling, the framework was found to be complete, easy to use, and useful, but not reliable enough to be readily applied in practice [MSBS02]. This conforms to the assessment by a number of researchers who consider SEQUAL too abstract and complicated to be applied by non-experts [MRvdA10, RMR10].

Guidelines of Modeling (GoM)

The Guidelines of Modeling (GoM) (ger. *Grundsätze ordnungsgemäßer Modellierung* [BRS95, BPV12]) are a set of general guidelines applicable for the creation of all types of conceptual models whose aim lies in ensuring a high quality of the modeling outcome [BRvU00]. Together with the SEQUAL framework, it was one of the first quality frameworks that could be applied to business process models, which underlines its importance for the BPM domain. In total, the GoM define six guidelines, formulated as the principles of correctness, relevance, economic efficiency, clarity, comparability, and systematic design [SR98]. Whereas the first three guidelines are characterized as necessary preconditions for model quality, the latter can be seen to be merely optional. In detail, the guidelines impose the following requirements [BRvU00]:

- **Correctness**. The principle of correctness requires a model to be both *syntactically* and *semantically* correct. Whereas the former is given if a model conforms to the rule of the employed modeling language, the latter requires a conformance of the structure and behavior of the model with the real world. Consequently, this principle is strongly related to the syntactic and semantic quality types of SEQUAL.

- **Relevance**. The principle of relevance states that a model shall not contain any elements that can be removed without the model losing some of its meaning to the user. Furthermore, it states that a relevant object system shall be represented using a relevant modeling technique.

- **Economic Efficiency**. The principle of economic efficiency postulates that all other principles should only be followed to the extent that the expected benefits outweigh the required effort [RSS01]. Consequently, this may for instance lead to decreases in correctness and clarity. Possible means of improving economic efficiency include the use of reference models, advanced modeling tools, and model re-use.

- **Clarity**. The principle of clarity states that a model should be understood by its intended user. This requires models to be readable, understandable, and useful—not just for modeling experts, but also for others. Thus, this principle is reflected in the empirical and pragmatic quality types of the SEQUAL framework.
- **Comparability**. The principle of comparability requires the consistent application of all guidelines across an entire modeling project to ensure that models remain comparable. This means, for instance, that the same naming conventions and layout rules should be used in all models and by all modelers.
- **Systematic Design**. The principle of systematic design postulates that business processes should not be modeled in isolation, but together with their relevant environment, including input and output data, organizational units, and resources. This requires a modeling method with well-defined relationships between different types of models, such as the Horus Method [SVOK12] or the Architecture of Integrated Information Systems (ARIS) approach [Sch00].

Besides these general guidelines, the GoM also includes more specific recommendations for particular modeling views (e. g., process modeling, data modeling, or enterprise architecture modeling) and modeling languages [BRvU00]. Thus, the framework exhibits a hierarchical structure in which quality can be considered at three different levels: all conceptual models, models for a particular view, and models in a particular language. Similarly to SEQUAL, some GoM literature discusses concrete modeling guidelines and conventions (for instance regarding the layout of models or naming conventions for model elements), but no concrete metrics allowing for a straightforward operationalization are defined [BRvU00, BPV12]. Therefore, a certain degree of modeling knowledge is required to use the GoM, and thus their usefulness for novice modelers is limited [MRvdA10].

Other Frameworks

In [RMR10], Reijers et al. present the “Simple, Integrates, Quality (SIQ)” [sic] framework whose premise is that other quality frameworks are too general and/or abstract. Its main aim is to provide concrete metrics, tools, and guidelines by design that facilitate the application of the framework in a retrospective (analyzing existing models) or a proactive (maintaining quality while modeling) fashion. SIQ distinguishes between syntactic, semantic, and pragmatic quality, and thus builds on the same foundations as SEQUAL, namely the work by Lindland et al. [LSS94]. To achieve its goals, the authors define “walls” of checking and ensuring around these quality types that provide specific tools for ensuring high levels of quality. These tools include, for instance, process simulation and the Seven Process Modeling Guidelines (7PMG).

The “Quality Marks, Metrics, and Measurement (3QM)” framework proposed by Overhage et al. [OBS12] aims to fulfill the requirements for quality frameworks specified by Moody [Moo05]. To that extent, it defines a hierarchical structure with the three quality characteristics syntactic, semantic, and pragmatic quality at the second-highest level below overall model quality. Thus, 3QM can be seen as a further framework inspired by SEQUAL. At the lower levels, 3QM defines 9 subcharcteristics (e. g., correctness, relevance, completeness, and flexibility for semantics) and a total number of 35 associated quality metrics with respective measurement procedures. The authors also suggest concrete weights across the entire framework hierarchy, which allows calculating a single, aggregated quality measure for any process model.

Lastly, in [NPGP12], Nelson et al. propose the Conceptual Model Quality Framework (CMQF) as a combination of SEQUAL and the Bunge–Wand–Weber ontological model of Information Systems [WW90]. Since the authors aim for comprehensiveness, the CMQF is very extensive: it distinguishes eight “quality cornerstones” (i. e., sets of statements about physical or cognitive artifacts), four “quality layers” (physical layer, knowledge layer, learning layer, and development layer), and 24 “quality types”. As a result, CMQF is not limited to the quality of modeling artifacts, but also considers, e. g., the quality of the learning that models enable or the quality of physical artifacts that are developed based on models. However, the downside of this expressiveness is that the framework is even more convoluted and abstract than SEQUAL, and thus its usefulness in real-world application scenarios is questionable.

2.6.3 Quality Guidelines

When trying to create high-quality models, modelers are faced with a variety of challenges [MRvdA10]. Firstly, despite the fact that many organizations have started to adopt specialized process modeling tools, this type of software rarely offers any functionality that assists with quality improvement. Secondly, while top-down frameworks such as those presented in the previous section allow gaining a more holistic understanding of the properties of a good model, they remain rather abstract and cannot be easily applied by novice modelers without a certain level of modeling expertise. Therefore, Mendling et al. emphasize the importance of providing concrete, straightforward guidelines grounded in empirical research that support modelers during the modeling process in an actionable fashion [MRvdA10]. Adherence to these guidelines can then be enforced (or at least increased) by extending modeling software with respective functionality that warns modelers about any violations. Such guidelines are common in other areas as well, and can for instance be found in the context of programming, where they are intended to help programmers with writing high-quality source code [KP78].

Seven Process Modeling Guidelines (7PMG)

One set of guidelines that satisfies the requirements of being grounded in empirical research as well as being concrete enough to be easily applicable are the "7PMG" proposed by Mendling et al.[MRvdA10, p. 130]: (1) Use as few elements in the model as possible. (2) Minimize the routing paths per element. (3) Use one start and one end event. (4) Model as structured as possible. (5) Avoid OR routing elements. (6) Use verb-object activity labels. (7) Decompose the model if it has more than 50 elements. This list of modeling rules was derived from numerous studies in which the authors investigated which quality metrics most strongly impact understandability and error probability. Thus, applying them grants modelers a high probability of creating understandable, error-free process models. Furthermore, the 7PMG can not only be applied when creating entirely new models, but are equally relevant for improving already-existing artifacts. Mendling et al. later refined the 7PMG with new insights, and now suggest 31 elements as a threshold for guidelines (1) and (7), at most 3 inputs and outputs per connector for guideline (2), and up to 2 start and 2 elements for guideline (3) [MSGGL12].

Other Guidelines

La Rosa et al. discuss further possibilities to deal with the complexity of business process models in [LtHW+11] and [LWM+11]. The first paper outlines modifications to the visual representation of a model, which the authors refer to as concrete syntax [LtHW+11]. The proposed operations include highlighting groups of elements with enclosures, applying changes to the primary and secondary notations of model elements, and providing layout and naming guidance. Conversely, the second article is concerned with changes to the formal specification of a model regardless of its visualization, which the authors call abstract syntax [LWM+11]. Here, the set of proposed modifications includes duplicating elements, modularizing models (e. g., using top-down refinements), merging, and removing elements. In both papers, the recommendations are presented as design patterns that form an extensive toolkit. However, most patterns remain rather abstract and lack clear guidance about when they should be applied, thus making them difficult to use without modeling expertise.

Another set of guidelines is presented by Weber et al. in [WRMR11]. In their work, the authors focus on the identification of refactoring opportunities (called process model "smells") in large process model repositories, and refactoring techniques through which models can be improved. The former includes issues such as unclear model and activity names, a high model complexity, and redundant process fragments. In turn, the latter consists of, e. g., renaming activities, substituting process fragments, and removing redundancies. Together, the combination of refactoring opportunities and techniques could be used to formulate concrete modeling guidelines, although the authors do not present them in such a way.

In [vdA12b], van der Aalst employs the experience gained from over 100 process mining projects to compile a list of seven common problems related to process modeling. These include, for instance, "aiming for one model that suits all purposes" [vdA12b, p. 563], "using static hierarchical decomposition as the only abstraction mechanism" [vdA12b, p. 565], and "color, size, and location without meaning" [vdA12b, p. 567]. While the author further deliberates on each of these issues, the discussion eventually remains too abstract so that concrete modeling guidelines can only be derived with additional effort.

Finally, Moreno-Montes de Oca and Snoeck conduct an extensive review of process modeling guidelines proposed in academic literature and present a list of 27 concrete recommendations grouped into the categories "counting elements", "morphology", and "presentation" [MS14]. Compared to the 7PMG, the authors present a broader set of rules that also cover aspects such as the secondary notation. Furthermore, unlike some of the remaining publications, the proposed guidelines are concrete enough to be applied with relative ease in practice.

2.7 Business Process Model Quality Metrics

No single quality metric can account for all quality aspects of a business process model [Men08]. Consequently, this section presents a variety of quality metrics grouped according to the nature of the construct that they measure. In particular, these are metrics for *planar variables* (i. e., element position) in Section 2.7.1, metrics for *retinal variables* (i. e., visual variables) in Section 2.7.2, metrics for *complexity* in Section 2.7.3, metrics for *textual model contents* in Section 2.7.4, and lastly metrics for *model semantics* in Section 2.7.5. The description of each metric includes its basic idea and underlying rationale, its expected impact on model quality and empirical results discussing these claims, and information about how it can be computed. In particular, the following details are provided for all metrics:

Measurement details

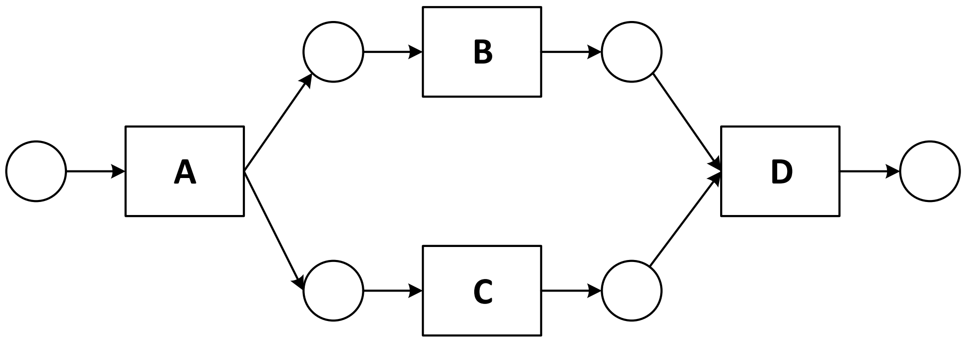
- *Absolute measurement*: Specification of an absolute measurement procedure $a(G)$. Note that the absolute measurement may in some cases already lie in the interval $[0, 1]$, in particular for metrics with a suggested upper bound smaller than 1.
- *Optimization goal*: minimization or maximization. In the following, it is assumed that the relative measurement procedure $r_i = s_i \circ a_i$ considers the optimization goal and the bounds described below to ensure that its final value is in the interval $[0, 1]$.
- *Lower bound*: The smallest possible value that $a(G)$ can assume. If the goal is minimization and the absolute measurement for a concrete model is smaller than or equal to the lower bound, its relative measurement procedure yields a perfect quality of 1. Inversely, if the goal is maximization and

the absolute measurement for a concrete model is smaller than or equal to the lower bound, its relative measurement procedure yields the lowest quality of 0. As most quality metrics have a lower bound of 0, this will be assumed as the standard in the following and only divergent values will be noted explicitly.
- *Upper bound*: The largest possible value that $a(G)$ can assume. If the goal is minimization and the absolute measurement for a concrete model is $\geq$ the upper bound, its relative measurement procedure yields the lowest-possible quality of 0. Inversely, if the goal is maximization and the absolute measurement for a concrete model is $\geq$ the upper bound, its relative measurement procedure yields a perfect quality of 1.

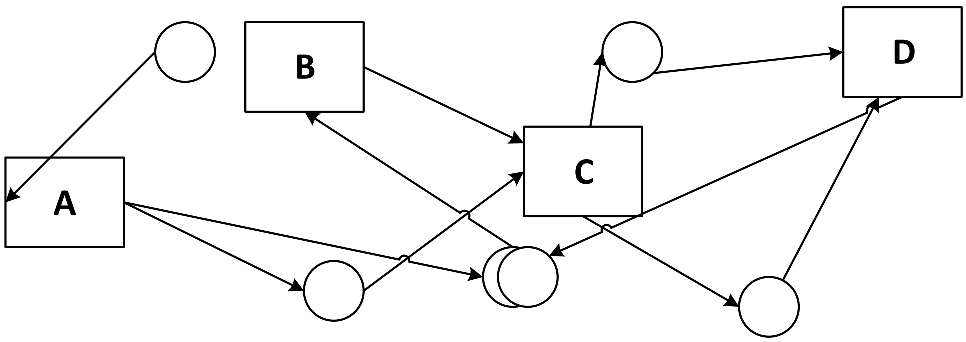
Note that there exists a vast amount of quality metrics, and the aim of this section is not to provide a comprehensive enumeration, but to illustrate the most common and relevant metrics that can be found in academic publications. In particular, the following two aspects of quality will be disregarded: *syntactic quality*, as this issue is comparably trivial and many modeling tools prevent syntactic errors, and the verification of process models via *formal properties* such as deadlock-freeness, liveness, boundedness, soundness, or well-structuredness, as they require process modeling languages with well-defined execution semantics and the described metrics are intended to be language-neutral. For additional information on the latter, the reader is referred to relevant literature on Petri nets, such as [vdA98, vdAvH02, GV03, vdAS11, vdAvHtH^{+}11].

2.7.1 Planar Variables

As with any other visual depiction of information, a clear distinction between a business process model and its representation can be made, and for any business process model, an infinite number of possible drawings exists. This is a result of the fact that modeling notations consist of two parts—the primary and the secondary notation—for which an infinite number of possible configurations exists. As a comparison between Figure 2.13a and Figure 2.13b reveals, not all possible representations are equally useful; while a good drawing may help the modeler to gain a better understanding of the depicted process, a poor visualization could even lead to false conclusions (cf. [DETT99]). Arguably,



(a) Good configuration.



(b) Bad configuration.

Figure 2.13: Quality metrics example: Planar variables.

the most important part of drawing a process model consists of finding appropriate positions for its elements that result in a drawing that can convey the structure and meaning of the underlying process as well as possible [DETT94]. Assuming a visualization in two-dimensional space, the respective variables to be determined—namely horizontal and vertical position—are also referred to as *planar variables* [Ber67].

In graph drawing literature, the notion of quality is usually referred to as *graph drawing aesthetics*, and regarding element positioning, a broad range of different metrics has been proposed (see, e. g., [DETT94, DETT99, Pur02b, BRSG07]). These metrics often conflict with one another so that improving one may result in a decrease of the other [DETT99]. Consequently, the extent of the impact of particular metrics on human understanding is of particular interest, as knowing on which of two conflicting metrics to focus greatly facilitates making design decisions. For this purpose, implementers may refer to a large body of work aiming to evaluate and rank metrics for general graph drawings, in particular the series of experiments by Purchase et al. [PCJ95, Pur97, PCJ97, Pur00, Pur02a, PPP12]). However, it has been noted that what may be true for general graph drawings does not necessarily have to be equally useful for domain-specific visualizations [PMCC01, BRSG07]. Thus, as comparable research exists for business process models only to a far lesser extent—one notable exception being the study conducted by Effinger et al. for BPMN [EJS10]—any indications made here about the relative importance of individual metrics must be regarded with a certain degree of caution.

Edge Crossings Already in 1953, Moreno, a pioneer in the field of social network analysis and visualization, remarked that "[a] readable sociogram is a good sociogram. To be readable, the number of lines crossing must be minimized. The smaller the number of lines that are crossing, the better the sociogram" [Mor78, p. 141]. The underlying rationale of this metric is that crossing lines make it more difficult to trace paths between nodes, thereby reducing the understandability of a diagram. Since then, edge crossings have become one of the most widely-discussed aspects of graph drawing aesthetics and are mentioned in almost all general deliberations on the topic (e. g., [DETT99, Pur02b]). This may be due to the fact that the concept is very simple to explain and its impacts easy to understand, as a brief examination of Figure 2.13 quickly reveals. Edge crossings have consistently been found to be the most important metric based on planar variables for understanding general graph drawings [PCJ95, Pur97], UML class diagrams [Pur02a], and BPMN diagrams [EJS10].

Measurement details

- *Absolute measurement*: Total number of pairwise intersections of edges in G.
- *Optimization goal*: Minimization.
- *Lower bound*: Crossing number of G; the computation of which is an NP-complete problem [GJ83]. 0 if G is planar; 0 may be used as a substitute possibly leading to an underestimation of $r(G)$.
- *Upper bound*: Maximum number of possible edge crossings; computed as the difference between the total number of edge pairs and the number of impossible crossings in a straight-line drawing [Pur02b].

Edge Bends While edges are most commonly drawn as straight lines, this does not always have to be the case. Introducing so-called *bend points* allows manually influencing how edges are routed across the visual canvas [CVR+12]. The latter can for instance be used to reduce the number of edges with crossings at the expense of increasing their lengths and making them more difficult to follow. Minimizing the number of edge bends in a graph drawing has been discussed as an aesthetic criterion as early as 1987 [Tam87]. Furthermore, it is consistently mentioned as one of the most important criteria influencing the human understanding of graph drawings—typically second to edge crossings [PCJ95, PCJ97, Pur02a]. When computing certain other metrics, it is important to also take bend points and individual line segments into accounts. For instance, adding a bend point separates an edge into two arcs, thereby increasing the number of possible crossings. Thus, the computation of this metric requires a prior *bends promotion*, i. e., the creation of a new graph in which bend points and the edges they create are added to G [Pur02b]. The maximum number of bends exhibited by a single edge in a drawing can also be used as an alternative metric called *curve complexity* [ACD+10, DEL09, DDLM10].

Measurement details

- *Absolute measurement*: Total number of edge bends.
- *Optimization goal*: Minimization.
- *Upper bound*: Unlimited; threshold value needed but not provided by literature. A reasonable threshold could be computed as $b * |A|$, with $|A|$ being the number of arcs in G, and b a constant specifying the average number of allowed bends per arc.

Node Occlusion The phenomenon of overlapping nodes in a graph drawing is called *node occlusion* or *node overlapping* [DS09]. This prevents readers from gaining a full understanding of the contents of a model and may thus yield incorrect conclusions. For instance, the arrangement of nodes in Figure 2.13b makes it difficult to identify the terminal node of the depicted process. Dunne and Shneiderman note that this simple metric is often ignored in recent literature, which may be due to the fact that it is perceived as so trivial that it merits no further discussion [DS09]. Due to the small size of many real-world business process models (cf., for instance, [MMN+06]), it can be presumed that creating a drawing without node occlusion should be possible in most cases. Node overlaps have been confirmed as an important metric in a study using BPMN, albeit to a lesser extent than, e. g., edge crossings [EJS10]. A strongly-related metric that may be used in addition to node occlusion is *edge tunnels* [DS09]. An edge tunnel occurs when a node overlaps with a connection, thereby visually dividing it into two separate edges, which may again imply incorrect execution semantics.

Measurement details

- *Absolute measurement*: Number of nodes involved in at least one overlap [DS09].
- *Optimization goal*: Minimization.
- *Upper bound*: Total number of nodes in G, although the use of a smaller threshold value may be reasonable.

Crossing Resolution The term *crossing resolution* can formally be defined as “the smallest angle formed by a pair of crossing edges” [ABS11, p. 62]. For instance, whereas the left-most two edges of Figure 2.14 intersect at about 40 degrees, the right-most two edges are orthogonal, thereby crossing at 90 degrees. If the resolution of all edge crossings is 90 degrees, the corresponding drawing is also referred to as a *right angle crossing drawing* [ACD$^+$10, DEL09]. Through a series of experiments, Huang et al. determined that the negative impact of edge crossings on readability becomes negligible if their intersection angles are at least 70 degrees [Hua07, HHE08]. Therefore, increasing the crossing resolution of a business process model is a valid alternative to removing edge crossings (especially if the latter is infeasible), although a trade-off certainly has to be found.

Measurement details

- *Absolute measurement*: Sum of the deviation of the crossing resolution of all edges from the optimum of 90 (or 70) degrees.
- *Optimization goal*: Minimization.
- *Upper bound*: Number of edge crossings multiplied by 90 (or 70).

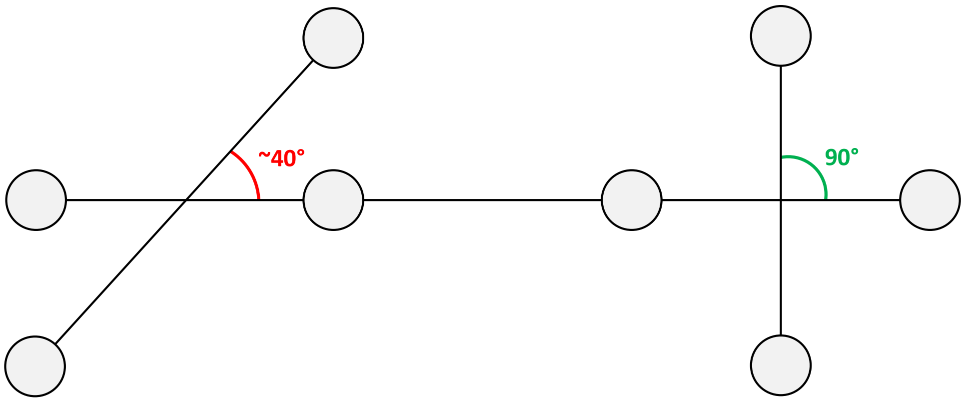


Figure 2.14: Quality metrics example: Crossing resolution.

Angular Resolution Not to be confused with crossing resolution, the metric *angular resolution* was first proposed by [FHH+93] and denotes the "smallest angle formed by two adjacent edges incident to a common node" [ABS11, p. 62]. Consequently, the maximum angular resolution of a given node in a business process graph can be calculated as the quotient of dividing 360 (degrees) by its number of incident edges. For instance, in Figure 2.15, the nodes A and B both have four incident edges, and thus their optimal angular resolution is $360/4 = 90$ degrees. Wheres node A deviates from optimal resolution by 60 degrees, this is not the case for node B around which its incident edges are spread out evenly. Maximizing angular resolution is desirable as it allows the reader to distinguish between different edges more easily, thereby facilitating path-following tasks. However, empirical studies have yet to research the impacts of angular resolution systematicall. Improving this metric comes at the cost of space, as a good angular resolution increases the area required by a drawing [GM99].

Measurement details

- *Absolute measurement*: Sum of the deviation of the angular resolution of all nodes from their respective optimum [Pur02b].
- *Optimization goal*: Minimization.
- *Lower bound*: 0, although more specific bounds exist for some graphs [FHH+93, GT94].
- *Upper bound*: Sum of the optimum angular resolution of all nodes.

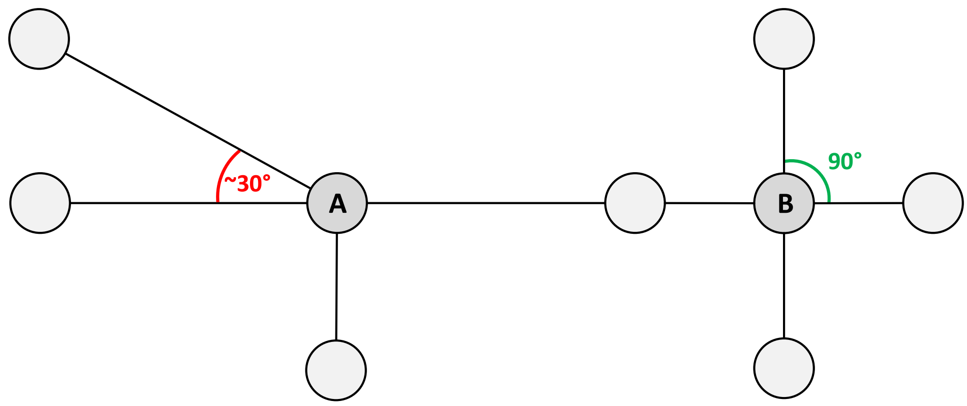


Figure 2.15: Quality metrics example: Angular resolution.

Consistent Flow Direction Generally speaking, the drawing of a directed graph can be said to have consistent (or uniform) flow if all of its edges point in the same direction, e. g., from left to right [Wad01]. For instance, whereas the flow of the process depicted in Figure 2.13a is consistent with regard to the direction left-to-right, this is not the case for the direction top-to-bottom, as two edges are pointing upwards. In business process models, the most common flow direction is left-to-right, although top-to-bottom is not uncommon as well [FS14, FS15]. Maintaining a consistent flow direction has been discussed as a quality metric for general graphs [Wad01], UML diagrams [PMCC01, ES09] and business process models [FS14, FS15], and from a theoretical point of view, a left-to-right flow can be expected to be most beneficial for human understanding as it corresponds to the most common text reading direction in Western cultures [FS14]. However, to date, no empirical study was able to detect any statistically significant effect of consistent flow [PMCC01, FS15].

Measurement details

- *Absolute measurement*: Number of edges with a particular flow direction.
- *Optimization goal*: Maximization.
- *Upper bound*: Total number of edges in G.
- *Remarks*: Requires the specification of a desired direction.

Orthogonality In an orthogonal drawing, all edges are drawn as horizontal or vertical lines–or possibly multiple alternating horizontal and vertical

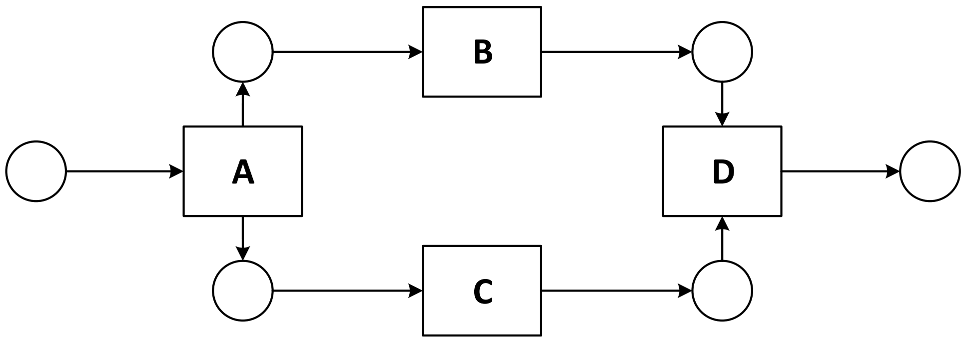


Figure 2.16: Quality metrics example: Orthogonality.

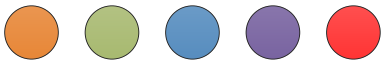
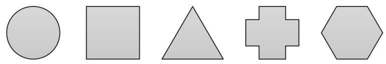
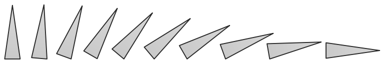
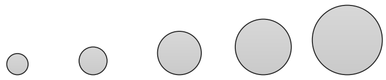
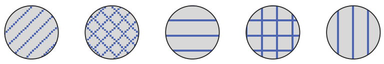
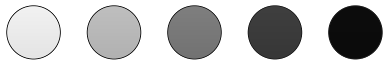
line segments if bend points are used [Tam87, PT00]. This is illustrated in Figure 2.16, which is an orthogonalized version of Figure 2.13a. Besides this standard definition of orthogonality centered around edges, PURCHASE also distinguishes an alternative interpretation called *node orthogonality* [Pur02b]. The main difference between both specifications is that in the latter, nodes must be placed on the intersection points of a predefined grid, but no assumptions about the orientations of edges is made. To date, empirical studies have neither managed to prove a significant impact of orthogonal layouts on human understanding for general graphs [Pur97, Pur02a] and UML diagrams [PMCC01], nor a preference of model users for this kind of arrangement [PPP12].

Measurement details

- *Absolute measurement*: Number of orthogonal (i. e., horizontal or vertical) edges. If more precision is required, the total angular deviation of edges from the closest horizontal or vertical gridline can also be computed [Pur02b].
- *Optimization goal*: Maximization.
- *Upper bound*: Number of edges in G.

Other Metrics Besides the metrics presented in the previous paragraphs, graph drawing literature also provides other aesthetic criteria that will only be briefly mentioned in the following. These criteria include the following [DETT94, CS96, Pur02b, BRSG07]: 1. The total area occupied by a drawing should be minimized. 2. The total length of the edges should be minimized.

Table 2.1: Bertin's visual variables. Source: based on [Ber67, Maz09].

Visual Variable		Visual Variable	
Hue		*Mark*	
Orientation		*Size*	
Texture		*Intensity*	

3. The maximum length of any edge should be minimized. 4. The lengths of all edges should be as uniform as possible. 5. The maximum number of bends exhibited by any single edge should be minimized. 6. The number of edge bends of all edges should be as uniform as possible. 7. The aspect ratio of the drawing should adhere to common screen aspect ratios. 8. Any symmetries existing in the graph should be emphasized.

2.7.2 Retinal Variables

Visualizing a business process model can be understood as the process of finding a visual mapping of all process elements to certain graphical attributes that specify all information that is required to create a visual representation of the former (cf. [CMS99]). These graphical attributes consist of a *spatial substrate* (axes and positions), *symbols* (also called *marks*), and additional *retinal variables* (also called *graphical properties*), namely size, (color) intensity, hue, orientation, and texture [CMS99, Maz09]. These attributes (without position) are summarized in Table 2.1 and were first described by Bertin in 1967, who referred to them as *visual variables* [Ber67]. An example illustrating the use of such attributes is depicted in Figure 2.17 as a modified version of Figure 2.13a. Here, the visual mapping has been enriched with the following information: red as a background color for all places, blue as a background color for all transitions, special marks for the split and join transitions A and D, decreased size of the transitions B and C, and lastly increased border size of the start and end places of the process.

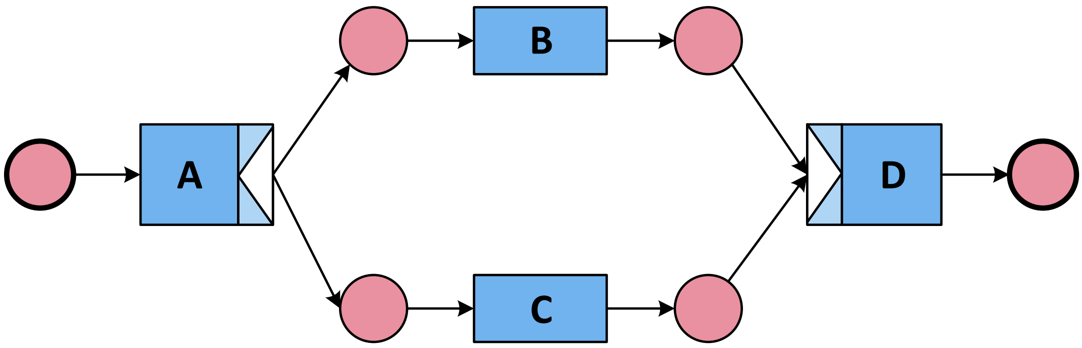


Figure 2.17: Quality metrics example: Retinal variables.

The purpose of visual mapping lies in finding a configuration of visual variables that supports the reader of a drawing—in this case a business process model—in whichever task they are carrying out [Maz09]. In this context, retinal variables play an especially important role, as they allow leveraging the visual capabilities of the human brain to integrate secondary information about a business process (e. g., cost, time, quality) into its visual representation without the loss of any information. However, it should be noted that more is not always better, and an excessive use of graphical properties may result in a visual information overload. Therefore, Ware suggests restricting the number of different hues to eight, orientations to four, sizes to four, and all other remaining attributes to ten [War04]. The creation of good visual representations is one of the key concerns of the *information visualization* discipline, and thus extensive guidance can be found in relevant textbooks [CMS99, War04, Maz09]. However, in BPM literature, retinal variables have been discussed to a far lesser extent, especially on the level of individual models. One exception is the study by Effinger et al., which has examined the impacts of element color and size [EJS10]. Some more work can be found on the level of modeling languages, for instance regarding the effectiveness of the design of routing symbols in different notations [FMSR10, LtHW$^+$11]. Due to the role of *position* as the most important graphical attribute [Mac88], planar variables have already been separately discussed in the previous subsection, and will be disregarded in the following. Furthermore, as the notations of all common business process modeling languages specify their own sets of notational symbols, the choice of graphical marks will also not be discussed as a quality criterion.

Syntax Highlighting In source code editors, syntax highlighting—the use of color to emphasize certain keywords in a meaningful way—has become a common practice that helps programmers to make sense of their code [RFME11, Sar15]. Despite the cognitive effectiveness of color, it is rarely used in notations for conceptual modeling [Moo09]. One exception are EPCs, where background colors are used to distinguish between different types of elements: green for functions, and purple for events [LtHW⁺11]. Based on these observations, Reijers et al. suggest adapting syntax highlighting to business process models by searching for pairs of matching connectors (i. e., the split connector at the beginning of a structured block and the corresponding join connector at its end) and highlighting them with the same background color [RFME11]. By using a different color for each pair, the model is visually subdivided into different components, thereby presumably improving the understandability of the model. Through an experiment, this was confirmed for novice modelers but not experts [RFME11], thereby mirroring the fact that syntax highlighting of source code is also most effective for more inexperienced programmers [Sar15]. It should be noted that the impact of colors on model understandability may vary for members of different cultures due to culture-specific preferences and characteristics. Specifically, members of the Confucian culture (e. g., Chinese) seem to profit from the use of colors in process models more than modelers with Germanic backgrounds [KRM16]. Finally, while Reijers et al. do not propose a concrete *syntax highlighting* quality metric, one can trivially be defined as the fraction of all matching connector pairs for which a distinct background color has been defined. Clearly, the concept of syntax highlighting could also be extended to other aspects of process models.

Measurement details

- *Absolute measurement*: Number of matching connector pairs in G with a distinct background color.
- *Optimization goal*: Maximization.
- *Upper bound*: Number of matching connector pairs in G.

Label Styles In some domains, such as cartography, the visual design of textual label plays an important role for the understanding of visual representations. While this topic has only received negligible attention in the context of

BPM so far, Koschmider et al. aim to fill this gap by providing recommendations for label design based on theoretical considerations [KFS15]. The authors suggest the use of lowercase letters, sans-serif and non-bold fonts, left-aligned text, high-contrast color choices for text and background, and placing labels in close proximity to their reference elements. Using these recommendations, a *label style* metric can be defined by counting all elements in a process graph G that have a textual label that is designed as suggested, and relating this figure to the total number of elements with textual labels. To date, the impact of label styles on understandability has not yet been studied empirically in the context of BPM.

Measurement details

- *Absolute measurement*: Number of model elements $e \in E$ that have a textual label with a design following the recommendations specified in [KFS15].
- *Optimization goal*: Maximization.
- *Upper bound*: $|E^{act}_{Name}|$, with $Name$ being the labeling function for textual labels.

Activity Icons As outlined in Section 2.2, textual labels play an important role for process model understanding as they contain the actual semantics of the depicted process [Leo13]. Consequently, the true meaning of a model element is revealed to a model user when he reads and understands its associated label [MRR10a]. Mendling et al. argue that comprehensibility can be increased even further when labels are paired with graphical icons that complement the former [MRR10a]. To that extent, the authors analyze a sample set of process models and derive 25 generic verb classes for activity labels, such as *to create*, *to remove*, *to search*, *to send*, and *to assess*, and propose fitting icons for each. Assuming the existence of a classification function that can reliably determine the verb class of an activity label, a quality metric for *activity icons* can be computed by counting the number of all activity nodes in a process graph G whose icon fits the verb class of their textual labels, and relating this figure to the total number of activities in G. To date, the impact of activity icons on understandability has not yet been studied empirically.

Measurement details

- *Absolute measurement*: Number of activity nodes in G that have both a textual label and an icon and for which the assigned icon fits the verb class of the textual label.
- *Optimization goal*: Maximization.
- *Upper bound*: $|E_{Activity}|$, the number of activity nodes in G.

2.7.3 Complexity

While the previous sections have discussed concerns related to the visual representation of a business process, this section focuses on those aspects of quality that depend on the inherent complexity of the underlying process itself regardless of its drawing. Formally, the complexity of a business process model can be defined as "the degree to which [it] is difficult to analyze, understand or explain" [Men08, p. 130]. Alternative terms that can also be found in the BPM literature include *understandability* [MRC07, RM11, HFL12], *comprehensibility* [FL11, FL15], and *clarity* [BRvU00, MR07]. Here, the term complexity is used to avoid any confusion that might result from the differences between understandability/comprehensibility and understanding/comprehension respectively. Whereas the former can be understood as an inherent property of a model, the latter is something that may or may not happen when it is read by a modeler [HFL12, HFL14]. Therefore, understanding and comprehension also depend on the personal properties of the model user and cannot be assessed without human involvement. As the purpose of this section is to provide complexity metrics that can be computed based on a model alone, the latter perspective will be disregarded in the following.

Studying the impacts of business process model complexity is an important concern, as the intricacies of real-life processes are difficult to control, and thus complex processes will often result in complex models [CMNR06]. For instance, Figure 2.18 represents an extremal case in which process mining techniques were used to derive a process model from patient treatments in a Dutch hospital [vdA11]. Clearly, the resulting model is incomprehensible for human readers for any possible configuration of planar and retinal variables. Furthermore, a broad range of different studies has shown that more complex models are also

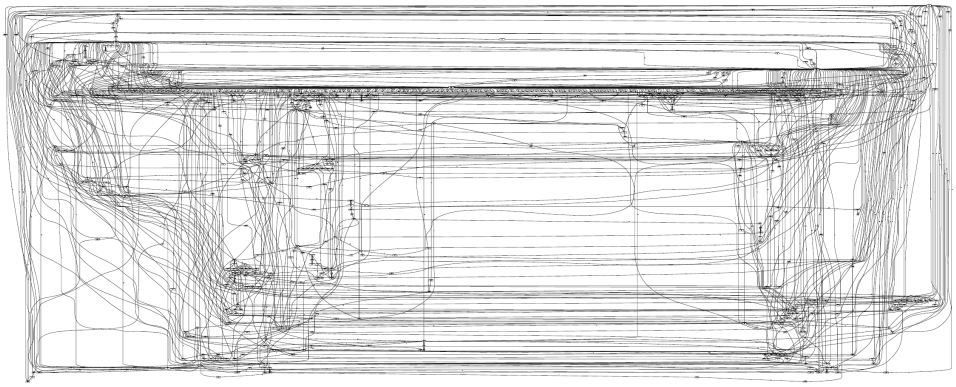


Figure 2.18: Quality metrics example: Complexity. Source: [vdA11, p. 302].

more likely to contain errors of various natures, such as syntax violations or deadlocks [MMN+06, MNvdA07, MVvD+08, Men08, Men09, RS15]. Regarding concrete metrics, initial proposals were formulated by adapting metrics from the context of software complexity to the area of business process models [CMNR06, GL06]. Building on this work, an extensive list of metrics has been compiled by Mendling with a focus on using them as predictors for the existence of model errors [Men08]. Finally, in a more recent report by Moreno-Montes de Oca and Snoeck, complexity metrics are operationalized by proposing actionable modeling guidelines for users to follow in an attempt to create more understandable models [MS14].

The metrics that are described in the following subsections are structured according to the groups used in [MSGGL12]: *size*, *connection*, *modularity*, *connector interplay*, and *complex behavior*. It should be noted that some metrics can be assigned to more than one group. In this case, they are presented in the subsection for which they hold the strongest relevance.

Size Measures

Metrics in this group relate to the size of a business process model as measured by the number of elements it contains or its length.

Size The size of a business process model is a rather simple quality metric that was first discussed for Petri nets [LY92, Mor99] and that can be understood as an adaption of the "lines of code" metric from software complexity

[CMNR06]. It is also a double-edged sword: if the model is too large, its understandability may suffer [MRC07]; inversely, with decreasing model size, the expressiveness of the model is also diminished. Additionally, empirical studies have connected large model sizes to the existence of formal modeling errors [MNvdA07], thereby further emphasizing the importance of finding "good" model sizes in practice. However, two challenges make this a difficult task. Firstly, different modeling languages vary in their verbosity, meaning that model sizes do not grow at the same rates. For instance, whereas both Petri nets and EPCs require certain elements to appear in alternation, this is not the case for BPMN. Therefore, it is easy to conceive a sample process for which its BPMN-based representation is much smaller. This issue may be addressed by defining multiple, language-specific quality metrics for size, or by relying on the smallest common denominator of all typical modeling languages, namely activities, as the singular determinant for this metric [CMNR06]. Secondly, there is no clear consensus on what a "good" size actually is. Sample guidelines propose keeping model sizes at 5 activities or below [MSGGL12], between 5 and 15 activities [KM96], or to decompose any process model with a total number of elements over 50 [MRvdA10]. As models with more than 50 elements have an error probability of over 50%, the latter should be seen as an extremal upper bound [MNvdA07]. To prevent the emergence of excessively large models, modelers can implement a top-down modeling approach in which business processes are first described at a very high level of abstraction and then successively refined with more detail by creating sub-process models for individual activities [SVOK12].

Measurement details

- *Absolute measurement*: Total number of nodes $|N|$ [Men08] or number of activities $|E_{activity}|$ [CMNR06]. Depending on the process modeling language, further specifications can be chosen as well [CMNR06].
- *Optimization goal*: Minimization.
- *Lower bound*: 0, or another reasonable bound > 0.
- *Upper bound*: Proposals include $|N| \leq 31$ [MSGGL12], $|N| < 50$ [MRvdA10], and $5 \leq |E_{activity}| \leq 15$ [KM96].

Diameter For a general graph, the diameter is defined as the length of the longest shortest path between of any of its nodes. Carried over to business process models, this is equivalent to the length of the longest, directed shortest path from any of its source nodes to any of its sinks. Thus, this metric has also been referred to as the *length* of a business process model [Nis98, Mor99]. Intuitively, the likelihood of a process model to contain an error increases with rising diameter, as it becomes more difficult for modelers to perceive the entire process at a single glance while modeling [Men08]. However, empirical studies have not been able to identify a significant impact of the diameter [MRC07, MS08b, RM11], and thus, it may presently be considered a metric of minor importance [Men08].

Measurement details

- *Absolute measurement*: $diam(G)$.
- *Optimization goal*: Minimization.
- *Lower bound*: 0, or another reasonable bound > 0.
- *Upper bound*: No recommendation is provided in relevant literature; however, a reasonable bound should be smaller than the upper bound for model size.

Connection Measures

Metrics in this group relate to the connections in a process model and their impact on the understandability of relationships between different model elements.

Density The density $\delta(G)$ of a business process graph G is the fraction of all possible edges in G that actually exist (cf. [LRJA10]). Consequently, the value of this metric can be computed as $\delta(G) = \frac{|A|}{|N|(|N|-1)}$ (note that for undirected graphs, every edge has to be counted twice). Empirical studies have shown that density plays an important role for the understandability of a business process model [MRC07, RM11]. The underlying rationale is that given two models with identical size but different densities, the model with lower density will be easier to comprehend since it contains a smaller number of arcs and thus has a simpler control flow.

Measurement details

- *Absolute measurement*: $\delta(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: $1/|N|$ for a business process consisting of a single sequence.
- *Upper bound*: Theoretically 1, although the syntax of some modeling languages may restrict the set of possible edges. It has been suggested that $\delta(G) < 0.033$ should be maintained to minimize error probability [MSGGL12]. However, as the number of possible arcs grows by the square of the number of nodes in G, this threshold may not be equally useful for all model sizes [Men08].

Coefficient of Connectivity The coefficient of connectivity CNC (also called coefficient of network complexity [CMNR06] or simply *average degree* [VCM$^+$07]) represents the average number of arcs per node in G [Men08]. Thus, it is strongly related to density and can be calculated as $CNC(G) = \frac{|A|}{|N|}$. Intuitively, a higher CNC can be associated with the same negative effects on understandability as a higher density. Furthermore, it has been shown that this metric has an impact on the error probability of a model [RS15].

Measurement details

- *Absolute measurement*: $CNC(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: $1 - \frac{1}{|N|}$ for a business process consisting of a single sequence.
- *Upper bound*: Theoretically $|N|-1$, although the syntax of some modeling languages may restrict the set of possible edges. A threshold of 1.021 has been suggested, although this value may be difficult to respect for very small models [MSGGL12].

Average Connector Degree Given the set C of all connector nodes in a business process graph G, the average connector degree ACD can be computed as the average number of incoming and outgoing arcs per connector, i. e., $ACD(G) = \frac{1}{|C|} \cdot \sum_{c \in C} deg(c)$ [Men08]. Evidently, the average connector degree is strongly related to the coefficient of connectivity and can thus be expected to impact understandability in a similar way. This has been confirmed by empirical studies that found this metric to be one of the most important complexity metrics [MRC07, RM11]. As an alternative or complementary metric, the *maximum connector degree* can be used; as the name suggests, the main difference in measurement between both is that rather than calculating the average degree of connectors, the largest degree of any connector in G is determined [Men08].

Measurement details

- *Absolute measurement*: $ACD(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: 0, although under reasonable assumptions (connectors are only used if they actually split or join the control flow, i. e., they have at least one input and two outputs, or two inputs and one output), a more appropriate lower bound would be 3.
- *Upper bound*: A threshold value of 3 has been proposed by [MSGGL12].

Cross-Connectivity The *cross-connectivity* metric was designed to measure the cognitive effort that is required to understand the relationship between any two nodes of a process model [VRM+08]. It is based on the idea of finding the most complicated path between any pair of nodes $n, m \in N$, and then assigning it a complexity value $V(n, m)$ based on its connectivity. To that extent, weights for all nodes and arcs in a business process graph are first computed. These weights encapsulate the "business logic" of the metric and depend, e. g., on the types of connector nodes and their degrees. Using these constructs, the cross-connectivity of G can be computed as the sum of the values of the connections between any pair of nodes divided by the total number of possible arcs, i. e., $CC(G) = \frac{\sum_{n,m \in N} V(n,m)}{|N| \cdot (|N|-1)}$. Empirical research indicates that process models with lower cross-connectivity tend to be less error-prone [VRM+08], and that together with other metrics, the former can be used as a predictor for the degree of understandability a business process model possesses [VRM+08, RM11].

Measurement details

- *Absolute measurement*: $CC(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Upper bound*: 1.

Modularity Measures

Metrics in this group relate to how well a business process model is structured into separate, distinct parts that a human reader can make sense of in isolation.

Separability In Section 2.5, the assumption that all business process graphs G are (at least) weakly connected was introduced. In such a graph, any node whose deletion would split G up into multiple connected components is called a *cut vertex*. The separability metric SEP counts the total number of cut vertices and puts them in relation to the total number of nodes (excluding sources and sinks which can never be cut vertices), i. e., $SEP(G) = \frac{n \in N | nisacutvertex}{|N| - |N_{start}| - |N_{end}|}$ [Men08]. A higher separability points to a more modular process whose individual parts can be considered in isolation and can thus be expected to correlate with easier understanding [MRC07]. However, while some studies were able to confirm this relationship [MS08b], others found no such effect [RM11]. In addition, separability was also confirmed as a relevant predictor for the error probability of a process model [Men08].

Measurement details

- *Absolute measurement*: $SEP(G)$ as defined above.
- *Optimization goal*: Maximization.
- *Upper bound*: 1, if G consists of a single sequence of activities. It has been suggested that $SEP(G) \geq 0.49$ should be maintained to minimize error probability [MSGGL12].

Sequentiality The sequentiality of a business process model can be defined as the extent to which its control flow consists of pure sequences of activities [RM11]. It can thus be determined as the fraction of arcs in G between non-connector nodes, as only connectors allow the control flow to deviate from being sequential [Men08]. Given the set of non-connector nodes $NC = N \setminus C$, the sequentiality value of G is calculated as $Seq(G) = \frac{|A \cap (NC \times NC)|)}{|A|}$. The rationale of this metric is that sequences represent the simplest possible type of control flow [Men08]. Therefore, it can reasonably be assumed that a highly sequential model will be easier to understand than a model of identical size with complex interrelationships between its activities [RM11]. Empirical research has found the sequentiality of a process model to be a relevant predictor for its error probability [RS15], but not for understandability [RM11].

Measurement details

- *Absolute measurement*: $Seq(G)$ as defined above.
- *Optimization goal*: Maximization.
- *Upper bound*: 1, if G consists of a single sequence of activities. It has been suggested that $Seq(G) \geq 0.21$ should be maintained to minimize error probability [MSGGL12].

Structuredness The basic idea of the *structuredness* metric is to examine which fraction of a process model consists of nested blocks with matching join and split connectors [Men08]. From an algorithmic point of view, structuredness is determined by applying a set of reduction rules to a given G to derive a *reduced* business process graph G' that consists of anything in G that is unstructured. The value of the metric is then calculated by putting the sizes of both graphs in relation, i. e., $Struc(G) = 1 - \frac{|N'|}{|N|}$, with N' being the set of nodes of G'. If a process model only consists of structured blocks, G' and hence N' will be empty, and thus $STR(G)$ will have a value of 1. While empirical research has not been able to confirm a relationship between structuredness and model understandability [MS08b, RM11], it has been shown that a higher structuredness indicates a lower error probability [MNvdA07, Men08, RS15].

Measurement details

- *Absolute measurement*: $Struc(G)$ as defined above.
- *Optimization goal*: Maximization.
- *Upper bound*: 1, if every node in G is part of a structured block. It has been suggested that $Struc(G) \geq 0.79$ minimizes error probability [MSGGL12].

Depth For structured business process graphs (i. e., models in which every split connector has a corresponding join and vice versa), the depth of a node $depth(n), n \in N$ is given by the maximum number of split connectors minus join connectors that lie on any path by which it can be reached from a source node. Consequently, the depth of G as a whole is determined by the maximum (or alternatively the average [GL06]) depth of any of its nodes, i. e., $Depth(G) = \max\{depth(n) \mid n \in N\}$. Thus, the notion of depths measures the extent to which G consists of nested, structured blocks [Men08]. However, Mendling also presents an algorithm for calculating the depth of unstructured process models [Men08]. A higher depth can be expected to correlate with a higher difficulty of understanding the respective model, as it indicates that more routing constructs must be considered while reasoning about the runtime behavior of a given activity [FL11]. However, such a relationship has not been confirmed so far [RM11]. Similarly, this metric does not seem to be a strong predictor for the error probability of a model [Men08].

Measurement details

- *Absolute measurement*: $Depth(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Upper bound*: Reasonably close to 0 and much smaller than $|N|$.

Connector Interplay Measures

Metrics in this group relate to the complexity of a business process model that results from the use of connector nodes influencing its control flow.

Connector Mismatch In a well-structured business process model, every split connector with a given out-degree has a corresponding join connector with identical in-degree [Men08]. However, most common modeling languages allow deviations from this, for instance by following a splitting *xor* with a joining *or*. In some cases, this may indicate a behavioral fault of the depicted process; for instance, an *xor* split followed by an *and* join results in a so-called deadlock–a state in which the process cannot be executed any further. Thus, avoiding connector mismatches is a reasonable guideline for process modeling [MRvdA10]. Based on [Men08], the mismatch metric can be calculated as $Mismatch(G) = \sum_{c \in Conn} \left(|\sum_{e \in E_{c_{split}}} deg(e) - \sum_{e \in E_{c_{join}}} deg(e)| \right)$. Here, for each connector archetype, the absolute difference between the degrees of the corresponding split and join nodes is summed up. For instance, if an *xor* split does not have a corresponding join, $Mismatch(G)$ is increased by the degree of the former. Connector mismatch can be used as a predictor for error probability [MNvdA07, Men08] and understandability [VRM$^+$08, RM11].

Measurement details

- *Absolute measurement*: $Mismatch(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Upper bound*: It has been suggested that $Mismatch(G) < 4.5$ should be maintained to minimize error probability [MSGGL12]. Less conservative thresholds of 9 [MNvdA07] and 6 [SGGRM12] have also been proposed.

Connector Heterogeneity The idea of this metric is that the more heterogeneous the connectors in a model are, the easier it becomes to introduce a connector mismatch, which may in turn point to a modeling error [Men08]. Thus, the connector heterogeneity shall assume a value of 0 if all connectors in a model have the same archetype (or there are no connectors at all), and 1 if the number of nodes is identical for all types. Based on the *information entropy* measure by Shannon [Sha48], the connector heterogeneity can be calculated as $CH(G) = -\sum_{c \in Conn} \left(p(c) \cdot \log_{|Conn|}(p(c)) \right)$. Here, $p(c) = |C_c|/|C|$ represents the fraction of all connector nodes with a specific archetype $c \in Conn$ and the base of the logarithm in the equation is given by the number of different connector archetypes that exist. While the impact of connector heterogeneity on understandability is small at best [MRC07, MS08b, RM11], this metric is an important predictor for the existence of model errors [MNvdA07, Men08].

Measurement details

- *Absolute measurement*: $CH(G)$ as defined above.
- *Optimization goal*: Minimize.
- *Upper bound*: 1, if all archetypes are represented equally. It has been suggested that maintaining $CH(G) < 0.4$ minimizes error probability [MSGGL12] and $CH(G) < 0.62$ maximizes understandability [SGGRM12].

Control Flow Complexity The *control flow complexity* of a business process model measures its behavioral complexity by accounting for the number of mental states that modelers have to keep track of as a result of the split connectors in the model [Car05]. This metric is based on McCabe's cyclomatic complexity [McC76] from the software engineering discipline, which was first applied to Petri nets in the 1990s [LY92, Mor99]. An adaption considering the specific semantics of business process models was later proposed [Car05] and validated [Car06] by Cardoso. In this work, the metric is defined as $CFC(G) = CFC_{and}(G) + CFC_{xor}(G) + CFC_{or}(G)$, where $CFC_{and}(G) = \sum_{s \in S_{and}} 1$, $CFC_{xor}(G) = \sum_{s \in S_{xor}} deg_{out}(s)$, and $CFC_{or}(G) = \sum_{s \in S_{or}} 2^{deg_{out}(s)} - 1$ represent the number of possible outcomes of *and*, *xor*, and *or* splits respectively. While the control flow complexity metric is easy to compute and understand, it was not found to be a good predictor for the existence of model errors [MNvdA07, Men08].

Measurement details

- *Absolute measurement*: $CFC(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: 0, if G contains only *and* connectors or no connectors.
- *Upper bound*: It has been suggested that $CFC(G) < 4$ should be maintained to minimize error probability [MSGGL12] and $CFC(G) < 13$ to maximize understandability [SGGRM12].

Gateway Complexity Indicator The *gateway complexity indicator* is a combined metric that is based on a linear combination of $Mismatch$, CH, CFC, ACD, the maximum connector degree MCD, and the total number of connectors TNC [SGGRM12]. Thus, it can be seen as a higher-level measure aiming to provide a summarized assessment of the complexity of a business process model that arises from its use of connector nodes. The gateway complexity indicator as proposed by Sánchez-González et al. is computed as $GCI(G) = 0.176 \cdot CFC(G) + 0.177 \cdot Mismatch(G) + 0.159 \cdot CH(G) + 0.175 \cdot ACD(G) + 0.180 \cdot MCD(G) + 0.179 \cdot TNC(G)$.

Measurement details

- *Absolute measurement*: $GCI(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Upper bound*: It has been suggested that $GCI(G) < 6.42$ should be maintained to maximize understandability [SGGRM12].

Complex Behavior Measures

Metrics in this group relate to how complex the behavior of a business process model is at runtime.

Cyclicity As described in Section 2.3, cycles which enable the repeated execution of parts of a business process are a common control flow pattern. The overall *cyclicity* of a business process graph can be calculated as the fraction of nodes in G that are part of at least one cycle, i. e., $Cyc(G) = |N_{Cyc}|/|N|$, where $N_{Cyc} = \{n \in N \mid n \rightsquigarrow n\}$ is the set of all nodes for which a path exists from themselves to themselves [Men08]. Intuitively, a cycle can be expected to have a negative impact on the understandability of a process model, as it represents a complex type of behavior that may potentially span a large part of the process. Furthermore, the metric was determined to be an important predictor for the probability of a process model to contain an error [Men08].

Measurement details

- *Absolute measurement*: $Cyc(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Upper bound*: It has been suggested that $Cyc(G) \leq 0.005$ should be maintained to minimize error probability, although this threshold is not reliable [MSGGL12]. Interestingly, this threshold requires a model with $|N| > 200$ given only a single node on a cycle, and thus it can reasonably be assumed that cycles should be avoided altogether.

Token Split The *token split* metric is an indicator of concurrency that measures how many new tokens (i. e., parallel threads of execution) can be created when a business process is enacted [Men08], thus signaling a higher degree of behavioral complexity [LY92]. It is dependent on the out-degrees of the *and* and *or* split connectors in a business process graph and can be calculated as $TS(G) = \sum_{s \in S_{and} \cup S_{or}} (deg_{out}(s) - 1)$. The subtraction of the constant 1 from each summand should be noted, as only *new* tokens are counted by $TS(G)$. Empirical research suggests that token split is neither an important predictor for the understandability of a process model [MS08b, RM11], nor for its likelihood to contain an error [MNvdA07, Men08]

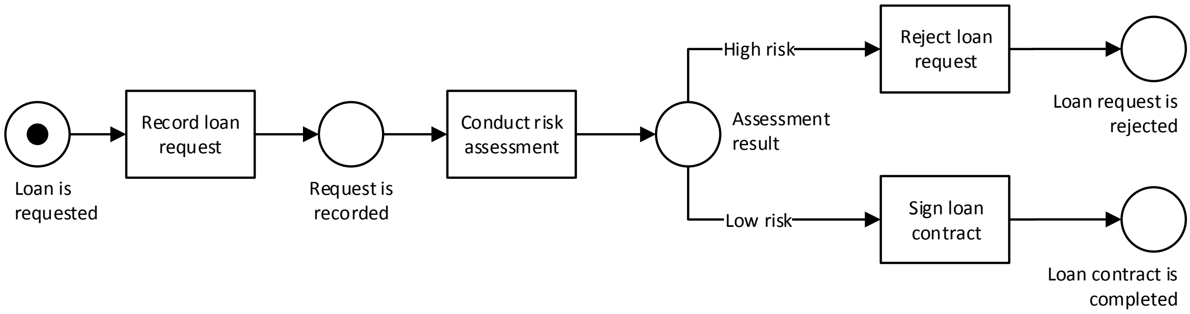
Measurement details

- *Absolute measurement*: $TS(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: 0, if G contains only *xor* connectors or no connectors.
- *Upper bound*: It was suggested that maintaining $TS(G) \leq 7$ minimizes error probability, although this threshold is not reliable [MSGGL12].

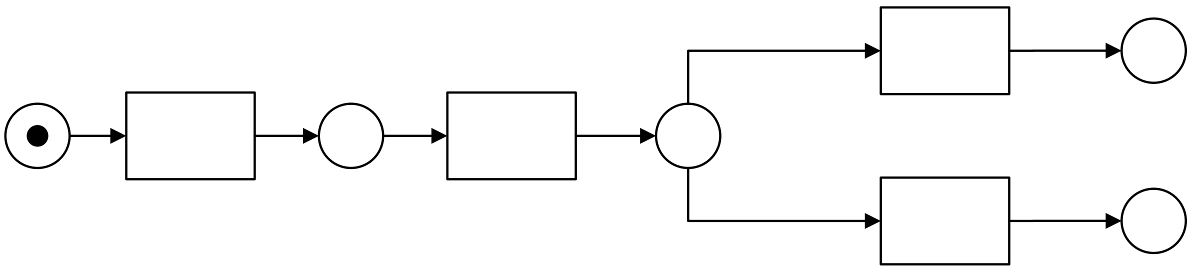
Sources and Sinks In an empirical study, Mendling et al. found a strong correlation between the number of source and sink nodes of a business process model and its probability to contain an error [MNvdA07]. On this basis, using only one source and one sink has been recommended as a pragmatic, easy-to-follow guideline for process modeling [MRvdA10]. This is even a syntactic requirement for certain types of process models, such as Petri-net based WorkFlow nets [vdA98]. Accordingly, the metric $SS(G)$ can simply be computed by counting the number of sinks and sources in G.

Measurement details

- *Absolute measurement*: $SS(G) = |N_{start}| + |N_{end}|$.
- *Optimization goal*: Minimization.
- *Lower bound*: 2.
- *Upper bound*: 2, although it has been suggested that maintaining $SS(G) \leq 4$ still results in a sufficiently low error probability [MSGGL12].



(a) Loan request as a Petri net with textual elements.



(b) Loan request as a Petri net without textual elements.

Figure 2.19: Quality metrics example: Textual elements. Source: based on [Men08, p. 9].

2.7.4 Textual Contents

Besides statements made according to the syntactic rules of a modeling language, process models typically also include textual content expressed using a natural language, such as activity and event labels or detailed descriptions. The importance of natural language for business process models can easily be explained by contrasting and comparing the two versions of the same process model that are shown in Figure 2.19: one with textual elements (Figure 2.19a) and one without (Figure 2.19b). By means of the names given to transitions and places, it can easily be inferred that the model represents a loan request process with two possible outcomes. Taking away these textual elements, little of the original meaning of the model is retained, leaving readers with the conclusion that a mostly sequential process with one *Xor* choice and two possible results is depicted. Thus, it can be concluded that natural language carries a significant portion of the semantics of any process model [Leo13].

As previously noted, the inherent ambiguity of natural language can make textual process elements difficult to interpret. This is especially true in large-scale, distributed modeling projects involving many different stakeholders with varying backgrounds and diverging sets of general and domain-specific vocabulary. Therefore, quantifying the quality of textual labels also plays an important role in maintaining process model quality. In this context, two broad types of quality metrics can be distinguished. Firstly, *syntactic* measures are related to desirable phrase structures for labels and whether models adhere to them. Secondly, *semantic* measures relate to the contents of labels and whether they can be easily understood and are free of ambiguities [LtHW$^+$11]. In the following, it is presumed that a label $Name \in \Lambda$ exists that can be assigned to model elements of all types. Furthermore, the terms "label" and "name" will henceforth refer to the aforementioned label $Name$.

Verb-Object Activity Labels One research stream in the area of textual model contents is focused on *activity labels*, which may have different styles as illustrated in Table 2.2, for instance *verb-object* or *action-noun*. In an empirical study, Mendling et al. examined the effects of different labeling styles on the perceived usefulness and ambiguity of the label and found *verb-object* style labels to be strongly superior in both regards independently of personal factors of the modelers [MRR10b]. Based on these insights, subsequent research has proposed various approaches to automatically detect the styles of labels [LSM09, LSM11] and refactor them to verb-object labels if needed [LSM10, LSM12]. A metric for the degree to which activity labels follow the verb-object style can be computed as $VOAL(G) = |E^{voal}_{Activity}|$, where $E^{voal}_{Activity} = \{e \in E_{Activity} \mid Name(e)$ *has verb-object style*$\}$ represents the set of all activities with a textual label with verb-object style. Analogous metrics can also be defined for other types of model elements such as events and gateways if their desired label structure is known (cf. [BDH$^+$09, DHLS09, LESM$^+$13]).

Measurement details

- *Absolute measurement*: $VOAL(G)$ as defined above.
- *Optimization goal*: Maximization.
- *Lower bound*: 0.
- *Upper bound*: $|E_{Activity} \cap E^{act}_{Name}|$, number of activities with a text label.

Table 2.2: Different activity labeling styles. Source: based on [LSM12].

Verb-object	Action-noun	Action-noun (of)	Action-noun (ing)
Create bill	Bill creation	Creation of bill	Creating bill

Naming Convention Adherence Due to the fact that the textual labels of process elements carry a significant portion of the semantics of a process model, their understandability is of considerate importance. However, a number of issues threaten their comprehensibility. For instance, distributed modeling projects involve a large variety of different modelers who may employ inconsistent labeling styles and vocabularies. This impedes the analysis, comparison, and integration of the affected process model collections [BDH+09, DHLS09]. Furthermore, the use of terms for which synonyms (different words with the same meaning) and homonyms (different meanings for the same word) exist introduces ambiguities into process models that may be difficult to resolve [DHLS09, PLM15]. Further understandability problems may result from typing errors [DHLS09], excessive text length [MS08b], and the use of uncommon words [KUHO15]. Based on such considerations, a number of approaches that aim to detect problems regarding the naming of process model elements and improving textual labels in various ways have been proposed, e. g., [BDH+09, DHLS09, MRR10a, KUHO15, PLM15]. The specifics of these proposals are not relevant for this thesis, and thus for the definition of a quality metric, the existence of a function measuring whether a textual label adheres to predefined naming conventions that result in a high understandability is assumed. This allows for the definition of an absolute measurement procedure $NCA(G) = |E^{nca}|$, where $E^{nca} = \{e \in E \mid Name(e)$ *adheres to naming conventions*$\}$ represents the set of all process elements with a textual label that follow the defined naming conventions.

Measurement details

- *Absolute measurement*: $NCA(G)$ as defined above.
- *Optimization goal*: Maximization.
- *Upper bound*: $|E^{act}_{Name}|$, the number of all process elements for which a textual label has been defined.

Text Complexity In an empirical study, MENDLING AND STREMBECK determined that the string length of textual elements have a significant negative impact on the understandability of a process model [MS08b]. This allows for the definition of a rather trivial quality metric $Length(G) = \frac{1}{E_{Name}^{act}} \sum_{e \in E} length(Name(e))$ calculating the average text length of all labels. Besides the simple length of a textual label, more elaborate text complexity metrics may be employed as well, for instance based on the number of words in a label [LSM09] or by adapting readability measures and equations [Fle48, Fry68, McL69]. While the idea of shortening textual labels to improve their understandability is reasonable, it should be noted that shorter does not mean better if the semantics of the element are not properly conveyed.

Measurement details

- *Absolute measurement*: $Length(G)$ as defined above.
- *Optimization goal*: Minimization.
- *Lower bound*: 0.
- *Upper bound*: Unlimited; threshold value needed but not provided by literature. A recommendation could be derived from a sample set of process models by calculating the average length of their labels.

2.7.5 Semantics

The semantic quality of a process model relates to the question whether it properly corresponds to the depicted domain, i. e., whether it is correct and complete [Kro16]. Compared to the quality concerns presented so far, semantic quality is more difficult (if not impossible) to assess automatically and will usually be determined based on the knowledge of model users, which yields the notion of *perceived* semantic quality. Due to this complexity, little work on concrete quality metrics for semantic quality can be found in academic literature. Based on the SEQUAL framework, metrics for completeness and validity are proposed below.

Completeness Following the SEQUAL framework, a business process model can be considered *complete* if it contains *all* relevant and correct statements about the represented domain (see Section 2.6.2). More formally, a quality metric for that purpose could be defined as $Comp(G) = 1 - \frac{(|D \setminus G|)}{|D|}$, where D represents the domain depicted in the model and the set operation $D \setminus G$ accurately determines which part of the domain is missing in the latter [Kro16]. Clearly, such a measurement is difficult—if not impossible—in a real-world scenario, and thus an alternative approach is required. For the 3QM framework, Overhage et al. define completeness as the extent to which “carriers of meaning that are required to communicate a real world excerpt” [OBS12, p. 233] are present. Such carriers of meaning include activities, flow conditions, organizational units, data objects, and textual labels. In terms of the business process graph definition provided in Section 2.5, labeling functions can be interpreted as carriers of meaning, and thus for any given label $\lambda \in \Lambda$, a graph G can be said to be complete if the set of elements that can be labeled with λ, but are not, is empty, i. e., $|E^{mis}_{\lambda}| = 0$.

Measurement details

- *Absolute measurement*: $|E^{mis}_{\lambda}|$.
- *Optimization goal*: Minimization.
- *Upper bound*: $|E^{pot}_{\lambda}|$, the number of elements that can be labeled with λ.

Validity Following the SEQUAL framework again, a business process model can be considered *valid* if it contains *only* relevant and correct statements about the represented domain (see Section 2.6.2). More formally, a quality metric for that purpose could be defined as $Val(G) = 1 - \frac{(|G \setminus D|)}{|G|}$, where the set operation $G \setminus D$ accurately determines which statements in the model are not part of the domain, and thus either irrelevant or incorrect [Kro16]. As before, such a measurement is difficult—if not impossible—in a real-world scenario, and thus an alternative approach is required. For the 3QM framework, Overhage et al. define validity as the extent to which “the meaning of the model elements is consistent to the real world excerpt that has to be depicted” [OBS12, p. 236]. While the authors also mention that a comparison between models and their underlying real-world excerpts must be performed to arrive at a measurement, the exact specifics of how this can be done are not documented. However, in [Rit10a] it is proposed that this may be achieved by asking modeling participants to assess the relevance and correctness of model elements and their associated labels. Thus, unlike the other metrics presented in this section, validity cannot be computed without human assistance. Accordingly, a quality metric for validity $Val(G)$ shall be informally defined as the fraction of all model elements and labels that a modeler has judged to be both relevant and correct. If responses are collected from m modelers, valid statements are summed up in the numerator, whereas the denominator is multiplied with m.

Measurement details

- *Absolute measurement*: Number of valid model elements and labels.
- *Optimization goal*: Maximization.
- *Upper bound*: Total number of model elements and labels.

3 Gamification

> "*I think some users of a text editing system would be challenged by having the system automatically maintain scores like typing speed or number of corrections made. If the text-editing task is boring or routine for the user, this challenge might increase the pleasure of using the system. (It would almost certainly* not *increase the pleasure of using the system, however, if such scores were used for surveillance by organizational superiors, however.)*" [Mal82, p. 66].
> (Thomas W. Malone discussing "gamification" already in 1982.)

Gamification is a relatively new phenomenon that aims to transfer insights gained from the development of computer- and video-games to other domains. To that extent, gamification typically consists of transferring individual elements of games and game design to non-game contexts [DDKN11]. The main purpose of this chapter is to provide a structured insight into those aspects of gamification that are most important for the application part of this dissertation. To that extent, Section 3.1 first address the basics of gamification, including its history in brief, various attempts to define the term, and its most important characteristics. Next, Section 3.2 presents important examples of gamified applications and gamification products to make the concept more tangible. Further research areas related to gamification are outlined in Section 3.3. This is followed by an overview of game design elements that can be used in the context of gamification in Section 3.4 and an outline of the theoretical foundations that are often used to explain how and why gamification works in Section 3.5. The chapter then ends with an examination of the most important points raised by critics of gamification in Section 3.6.

3.1 Basics

Discussion about gamification started as an industry-driven trend around 2008 and was picked up in academia briefly thereafter, with wide-spread publication starting around 2010 and gaining traction ever since [DDKN11, HKS14]. However, the practice of gamification is not entirely new, and its roots can be traced back throughout the last few centuries to areas such as religious practice, music and dance, lifestyle, learning, killing, and labor [Nel12, Fuc14]. Consequently, whether gamification actually demarcates a new phenomenon [DDKN11] or is merely a new term for concepts that have existed for a long time [BP13, BVW15] is subject of an ongoing debate. Nevertheless, it is clear that gamification builds upon a broad foundation of ideas from various domains, including game design, psychology [SA14, PT15], Human-Computer Interaction [DDKN11, SA15], and Information Systems (IS) design [BVW15].

Due to the novelty and explosive growth of gamification, no consensus about the meaning of the term has been reached yet. Rather, multiple competing definitions originating from both academia and practice can be found, a selection of the most important of which is presented in Table 3.1. Besides the definitions themselves, this list also indicates whether they are academic, i. e., published in a peer-reviewed, scientific outlet, or not, and provides an approximation of their adoption as measured by the number of citations. The most widely-accepted and -cited proposal of the term was introduced by Deterding et al., who define gamification as "the use of game design elements in non-game contexts" [DDKN11, p.10]. Using this definition as a starting point, the following paragraphs explain the basic idea of gamification while highlighting commonalities with and differences to other proposals.

Application area. Gamification can be applied to any context, scenario, or setting, as long as neither the starting point nor the result is a game [DDKN11]. While some authors use broad terms such as *non-game contexts* [DDKN11], *gameless objects* [YPW14], or simply *something* [23] to reflect the wide range of possible applications, others do not specify to what gamification can be applied at all [ZC11, 5]. Lastly, the proposals by Huotari and Hamari [HH12] and Werbach [Wer14] assume a more narrow perspective by defining gamification around *services* and *activities* respectively. While some authors claim that gamification is always digital [5], there is no inherent need for either

Table 3.1: List of definitions of the term gamification. Numbers of citations as indicated by Google Scholar on the July 11th, 2016.

Definition	**Acad.**	**Source**	**Citations**
The use of game design elements in non-game contexts	✔	[DDKN11]	2024
A process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation	✔	[HH12]	379
The process of making activities more game-like	✔	[Wer14]	33
A process that integrates game elements into gameless objects in order to have gameful characteristics	✔	[YPW14]	2
The process of game-thinking and game mechanics to engage users and solve problems	✖	[ZC11]	973
The use of game mechanics and experience design to digitally engage and motivate people to achieve their goals	✖	[5]	12
The process of adding games or gamelike elements to something (as a task) so as to encourage participation	✖	[23]	n.a.

the starting point or the result to be an app or any other kind of software [DDKN11]. In practice, gamification has been applied to a large variety of different contexts, including education, training, and learning, health, wellness, and fitness, work, sustainability, crowdsourcing, marketing, and research [HKS14, SF15b]. A selection of pivotal examples is presented in Section 3.2.

Process. All proposals agree that gamification is something “to be done”, with five explicitly defining the term as a *process* and the remaining two characterizing it as the *use* of elements from games. Indeed, the process character is even contained in the word “gamification” itself, which is composed of the word “game” and the suffix *-ification*,. The latter stems from Latin or old French and denotes “the *process* of becoming”. Thus, as Werbach appropriately points out, gamification consists of making something that is not a game become more game-like [Wer14], though without turning it into an actual game. Following the classic process definition by Davenport, this means that gamification consists of “a specific ordering of work activities across time and space”[Dav93, p. 5], has “clearly defined inputs and outputs”[Dav93, p. 5] and produces some value for its beneficiaries. From a high-level point of view, the inputs of this process consist of a *gameless object* [YPW14] (e. g., a service [HH12], an activity [Wer14], or a software application) and game design elements, whereas the output is a “gamifed” version of the former that allows for the emergence of *gameful* experiences [HH12]. As of yet, no consensus about the exact details of the gamification process has been reached, and given the creative nature of designing games, it can be questioned whether defining a universal reference process for gamification is even possible. An overview of different approaches to gamification is included in Section 3.4.5.

Repertoire. The ingredients that go into transforming a non-game object during the gamification process have been referred to by Deterding et al. as *game design elements* [DDKN11]. These elements can be defined at various levels of abstraction, ranging from very concrete (e. g., points, badges, and leaderboards) to highly abstract (e. g., using a play-centric approach to designing a gamified solution). This understanding addresses the whole breadth of what games are, what components they consist of, how they are presented to the player, and how they are designed. Consequently, carrying out gamification from this perspective entails not only adding components that can frequently be found in games to non-game contexts, but also applying methods that are used in game design and thinking like a game designer. In contrast, other authors assume a more restrictive point of view and define gamification around *game elements* [YPW14] or simply *game mechanics* [ZC11, 5], which only represent a subset of game design elements situated at the more concrete end of the spectrum. In practice, this narrower understanding is accepted and employed more widely, which may be due to the relative ease of simply adding very specific elements of games to other applications when compared to applying higher-level design elements. Consequently, the three most frequent game design elements used in gamification were consistently found to be *Points, Badges, Leaderboards (PBL)* across multiple studies [HKS14, SA15, MHK16]. A more comprehensive overview of game design elements that may be used in gamification is provided in Section 3.4

Product. As previously mentioned, the result of gamification process should be a "gamified" version of the input artifact that has gameful characteristics [YPW14] and affords gameful experiences [HH12] while not becoming a game itself [DDKN11]. This means, e. g., that the experience of using a gamified running app or an enterprise software application should share certain similarities with the experience that comes from playing a game, thus making it more satisfying, fulfilling, engaging, and productive [McG11]. This is not as straightforward as it seems, as playing a game is highly subjective, and thus the same system could be experienced as a game, a gamified solution, or even something entirely else depending on who is using it [DDKN11, HH12]. Consequently, this makes it difficult to define guidelines regarding "how many" game design elements have to be used for the result to be a gamified system. Furthermore, the definition of the term "game" itself is debatable and can influence the boundary between gamified systems and full games [SF15b]. In practice, this ambiguity can lead to confusion about whether a concrete example constitutes gamification or not. For instance, the application Foldit (University of Washington, 2008) [CKT+10] has not only been called a *game* [QB11], but also a *serious game* [DAJ11] and *gamification* [Det12].

Outcomes. Clearly, gamification is only a means to an end and is ultimately intended to serve a purpose other than pure entertainment [DDKN11]. Hamari et al. provide a very simple model that explains how and why the outcomes of successfully applying gamification materialize [HKS14]. Firstly, they interpret the game design elements gamification makes use of as *moti-*

vational affordances, i. e., interventions that give the gamified system certain properties capable of supporting the user's motivational needs [Zha08]. These affordances enable gameful experiences [HH12], which in turn yield *psychological outcomes* such as enjoyment, engagement, attitude changes and fun [HKS14]. This ultimately results in *behavioral outcomes* that depend on the context in which gamification is applied. For instance, for an online learning tool, these might include increased participation, increased quality of participation, improved learning outcomes, or faster learning [HKS14]. Within the definitions presented in Table 3.1, the intended outcomes of gamification are represented inconsistently. Most academic proposals make no concrete mention of either psychological or behavioral outcomes [DDKN11, Wer14], or remain rather vague by referring to gameful experiences [HH12] or gameful characteristics [YPW14]. In contrast, the remaining definitions from practice refer to concrete psychological outcomes such as engagement and motivation, and behavioral outcomes such as solving problems, achieving goals, and encouraging participation [ZC11, 5, 23]. This difference might result from the fact that explanations of gamification targeted at practitioners must provide more concrete examples of what can be achieved with such an approach, whereas academic proposals try to remain more open to account for the various ways in which games can be experienced.

To substantiate the basic idea of gamification, a concrete example is presented in Figure 3.1. Here, the activity to be gamified (i. e., the non-game context) is running, one of the most common starting points of gamification in the area of health and fitness [LWC$^+$14]. Under the assumption that the gamification endeavor is carried out by an enterprise manufacturing running equipment (cf. Section 3.2.1), typical goals of gamification would be increased sales for the gamification provider vis-a-vis running more and improving running skills for the users. A good strategy for the provider to achieve his goal would be to tie the gamified experience to vendor-specific products and services that are required for participation. To conduct gamification, the vendor can employ a broad range of elements from game design, including mechanics and other game components, game design and development processes, and tools such as prototyping and playtesting. This results in the value proposition for customers that running using the gamified products and services is a more enjoyable experience that yields a higher performance. The entire process is

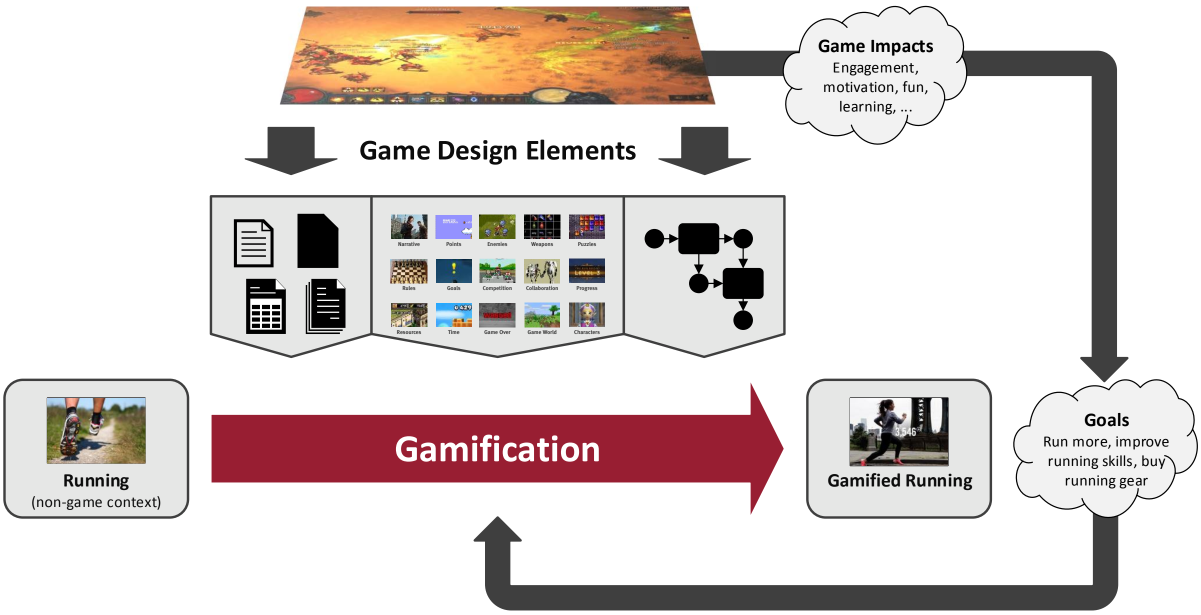


Figure 3.1: Basic idea of gamification illustrated by example of running.

based on the assumptions that such elements are responsible for the positive impacts associated with games, and that transferring them to other contexts allows creating gameful experiences.

Due to the rapid growth that gamification has experienced as a research area, a vast wealth of academic conference papers and increasingly also of journal articles discussing the topic can be found. Consequently, giving a comprehensive overview of gamification has become virtually impossible, and the increasing publication rate impedes efforts to remain up-to-date about recent insights. Therefore, as a first step towards "reconstructing the giant" [vBSN$^+$09] that gamification has become, the reader is referred to a number of unstructured as well as systematic literature reviews [WW02b, vBSN$^+$09, OS10] listed in Table 3.2 that have been conducted in the past few years to summarize and synthesize the state of the art. It should be noted that the listed reviews differ in focus, with some aiming to provide a general overview [SF15b], others focusing on particular application domains, including Information Systems ([BP13, SA15]), education [NZT$^+$14], software engineering [PGBP15], and crowdsourcing [MHK16], and finally some studies discussing the impacts [HKS14], theoretical foundations [SA14, PT15], and design approaches for gamification [MRGAM15].

Table 3.2: List of literature reviews on gamification.

Title	**Source**
A Descriptive Literature Review and Classification Framework for Gamification in Information Systems	[SA15]
A Literature Review of Gamification Design Frameworks	[MRGAM15]
Creating a Theory-Based Research Agenda for Gamification	[PT15]
Does Gamification Work? A Literature Review of Empirical Studies on Gamification	[HKS14]
Gamification - A New Phenomenon in Information Systems Research?	[BP13]
Gamification in Crowdsourcing: A Review	[MHK16]
Gamification in Software Engineering - A Systematic Mapping	[PGBP15]
Gamification in Theory and Action: A Survey	[SF15b]
Gamification of Education: A Review of Literature	[NZT+14]
Psychology Theories in Gamification: A Review of Information Systems Literature	[SA14]

3.2 Examples

Despite the fact that the term gamification is not an entirely new research area anymore, there is still a considerable amount of confusion about what gamification actually is. One major reason for this is that most definitions of the concept remain rather vague and are very open, thereby making it difficult to clearly distinguish between what is gamification and what is not. For instance, the most popular definition provided by Deterding et al. (cf. Section 3.1) exhibits vagueness in at least three aspects. Firstly, the term "use" it employs is very broad and thus neither helps with realizing, nor with identifying gamified applications. Secondly, the term "game design element" is equally vague and many of the most popular elements used in real-world gamification examples (i. e., points, badges, and leaderboards) are in fact not unique to games at all [32]. Lastly, the exact criteria for what constitutes a "non-game context" are unclear. While Deterding et al. equate this to any scenario that does not serve the primary purpose of enjoyment [DDKN11], Huotari and Hamari argue that this is highly dependent on the subjective perception of the individual player [HH12]. Due to the inherent vagueness of its various definitions, envisaging how gamification can look like when put into practice can be difficult. Thus, the aim of this section is to substantiate this by providing a cross-section of gamification examples in various fields. To that extent, Section 3.2.1 first presents some of the most popular, consumer-oriented gamification products that have helped shape public perception of the field. Afterwards, Section 3.2.2 will discuss further applications that have been presented in peer-reviewed, academic outlets. For examples about the corporate use of gamification and gamification platforms, the reader is referred to non-academic textbooks such as [ZC11], [Bur14], and [Her14].

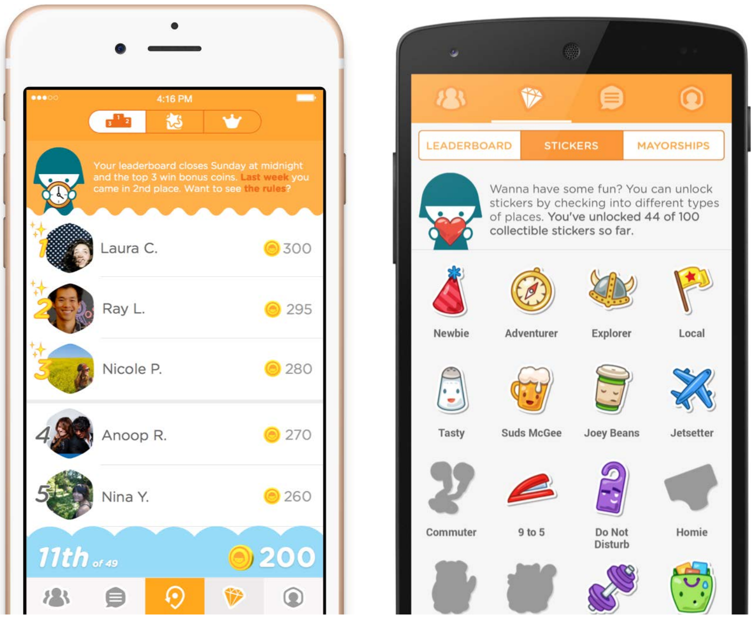
3.2.1 Practice

As outlined at the beginning of Chapter 3, gamification is an industry-driven trend. Consequently, many popular, consumer-oriented gamification products had already emerged before an academic debate on the topic was formed. To account for this, this section presents some of the most widely-known examples for gamification that originated in a non-academic context.

(a) Daily overview in the Nike+ Fuel app. Source: [24]. **(b)** Detailed information in the Nike+ Fuel app. Source: [24]. **(c)** Leaderboard in the Nike+ Move app. Source: [25].

Figure 3.2: Sample screenshots of the Nike+ Fuel and Nike+ Move apps.

Nike+ (Nike, 2006), a gamified fitness community based on various physical products and smartphone apps, is one of the earliest and most significant implementations of gamification. As such, it is often referenced as a motivating introductory example in literature on the topic (e. g., [ZC11, BP13, BL13, SA14]). The basic idea of Nike+ is the use of an activity tracker, the so-called *Nike+ FuelBand*, that measures the everyday physical activity of its wearer and converts it into a points score called *NikeFuel*. Users can set personal goals, track their progress, compare themselves with their friends on a leaderboard, and socially interact with other Nike+ users. Figure 3.2 show a few sample screenshots of the smartphone apps in the Nike+ ecosystem that illustrate its implementation of gamification. Firstly, Figure 3.2a presents the user with a high-level overview of their activity on a single day. Most notably, he can see the NikeFuel he has earned and how these gains are distributed over the day. Furthermore, a progress bar color-coded from red to green shows whether the user has already reached his personal goal. As seen in Figure 3.2b, more detailed information is also available, including the daily number of active minutes, comparisons



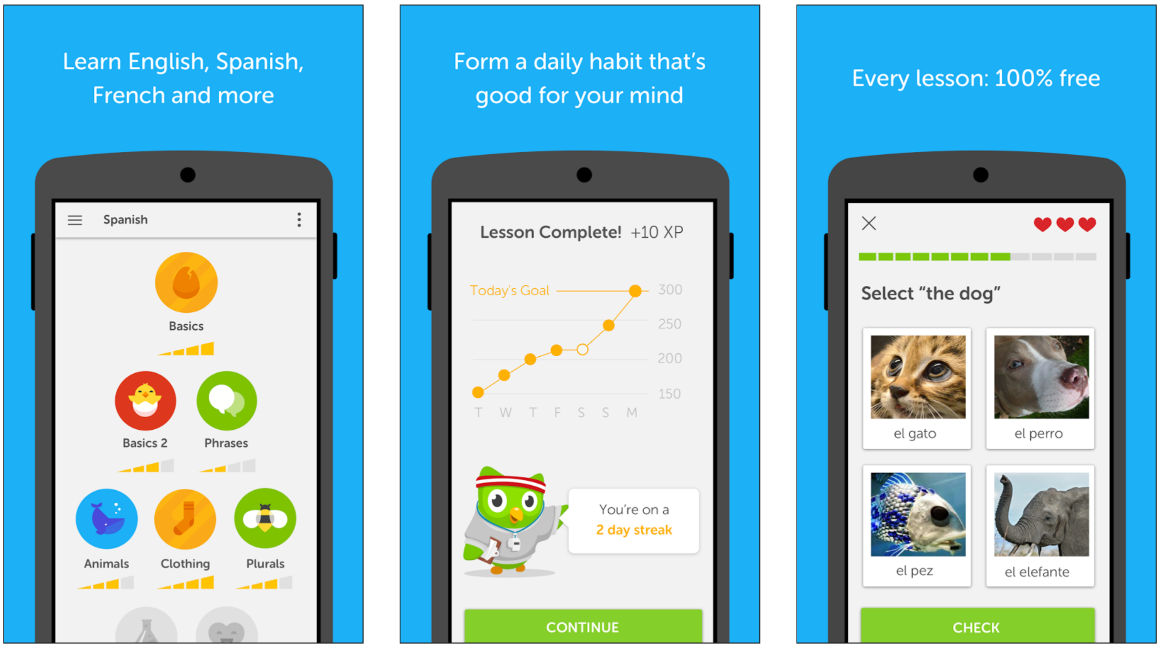
(a) Weekly leaderboard. **(b)** List of stickers.

Figure 3.3: Sample screenshots of the Swarm app. Source: [35].

with the user's own performance and the performance of others in the same age/gender group, and how often the user has reached his personal goals in a week. Lastly, Figure 3.2c depicts a leaderboard on which a user and his friends are ranked depending on the NikeFuel they have earned in a specific timeframe. Altogether, it can be seen that Nike+ provides numerous quantitative indicators for physical activity. It does this in a gameful manner by using game elements such as points scores, clear goals, challenges, and leaderboards, thereby turning sports into a gameful activity.

Foursquare (Foursquare Labs, 2009) is a local search and recommender service for facilities such as restaurants, cafés, and night clubs. Traditionally, the mobile application offered by Foursquare was built on gamification functionality; however, as part of a redesign, these features were transferred to a new companion app called *Swarm*[1]. In its core, Swarm is a social network that provides specific features based on the location data of its users. By using the app to check in at particular locations, users can earn virtual coins. The rewards increase under specific conditions, such as checking in on successive

[1] See: https://www.swarmapp.com. Last accessed: 2016-08-03.



(a) Overview of Spanish lessons. **(b)** Experience points and streaks. **(c)** Lesson with indicator for lives.

Figure 3.4: Sample screenshots of the Duolingo app. Source: [9].

days or checking in together with friends. Coins are then used to rank users on a weekly leaderboard, as illustrated by Figure 3.3a. Furthermore, users can earn various kinds of stickers (similar to achievements) such as those depicted in Figure 3.3b for checking in at different types of locations. The features provided by Swarm (and by Foursquare in the past) are representatives of rather shallow PBL gamification that can be applied to almost any type of non-game context and is comparably simple to implement, but typically only creates short-term rewards [WH12].

Duolingo (Duolingo, 2011) is a language-learning Web application with accompanying smartphone apps that employs gamification as a central strategy for its motivational fabric. It was created by a team around Luis von Ahn, an important figure of the *games with a purpose* movement, and was originally intended as a crowdsourced tool to "translate the Web into every major language" [vA13, p. 1]. With Duolingo, users can learn a wide variety of different languages such as Spanish, French, English, or German by applying their reading, listening, and speaking skills. Each language is subdivided into multiple "skills", i. e., sets of lessons that address different topics, including the basics of the language, animals, clothing, or food (cf. Figure 3.4a). By successfully completing lessons, users earn experience points that increase their proficiency level in a language (cf. Figure 3.4b). Furthermore, users can vary the difficulty level by setting daily experience goals for themselves which they must then try to beat. Doing so on subsequent days allows players to maintain a so-called "streak", which certifies their continuity in actively using Duolingo. Additional features include "lives" represented by heart icons that are lost by giving incorrect answers in lessons (cf. Figure 3.4c), competing with friends via experience-based leaderboards, and using a virtual currency called *lingots* to buy power-ups for lessons, visual upgrades, and further skills. The popularity of Duolingo, which has reached over 100 million users in 2015, indicates that learning is a fruitful application area for gamification [29].

3.2.2 Academia

This section presents examples for gamification that have been published in academic, peer-reviewed outlets. In contrast to Section 3.2.1, which is focused on popular applications, the aim of this section instead is to paint a more diverse picture of the field by illustrating more creative ways to apply gamification.

Multiple-choice quizzes. In [CCF13], the authors present Quick Quiz (Cheong et al., 2013), a gamified quiz software. Game elements provided by Quick Quiz include the possibility for players to earn points for answering questions correctly (see Figure 3.5a), time pressure generated by coupling the height of rewards with response speed, and a leaderboard enabling competition between different players. The authors used their tool in IT-based, undergraduate courses and examined its effects on students' engagement, enjoyment, and learning. Results indicate that students were engaged in a way that compelled them to want to complete the quiz, and that the majority of students believes that Quick Quiz improved their learning experience and outcomes.

Recycling. In [BAZN13], the authors present the emoticon-bin (Berengues et al., 2013), a gamified recycling bin for plastic bottles that rewards users with a happy smiley face (see Figure 3.5b) and a coin sound whenever a sensor detects that a bottle has been inserted. This positive reinforcement and feedback is intended to increase recycling rates, which are still very low in some parts of the world. Indeed, in a four-week experiment, the authors found the number of collected bottles to have tripled and determined a strong preference of users for the emoticon-bin over a standard recycling container. Furthermore, they demonstrate that the number of bottles that must be collected to justify the energy cost for operating the monitor of the emoticon-bin is negligible.

User authentication. In [EB16], the authors present Ariadne PathLogin (Ebbers and Brune, 2016), a gamified authentication method in which users need to complete a simple game to authenticate themselves instead of providing a password. The system requires users to choose a unique user name and a playing figure, and to define a path across a playing board that serves as the equivalent to the password. The latter is illustrated in Figure 3.5c, where red arrows indicate obvious possible movements, and blue arrows indicate "hidden" possible movements that can be chosen to confuse possible attackers. In the authentication phase, the user needs to provide the correct user name, figure, and path to gain access to a system. In a preliminary evaluation, the authors find the system to be feasible and usable in general, but note that it was considered to be rather time-consuming, which may make it most suitable for contexts requiring higher levels of security.

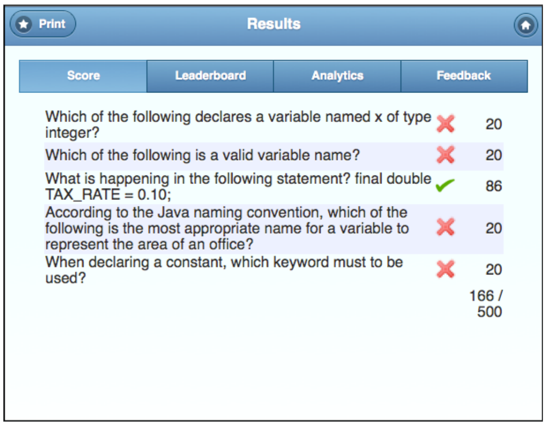
Job seeking. In [vdKK15], the authors present Kindle (van der Kruys and Khan, 2015), a gamification concept intended to motivate job seekers to look for work. To that extent, they describe an online system that allows users to upload their *curriculum vitae* (CV) as a central game object, the main goal of the platform being quality improvement. By giving feedback on the CVs of others, players can earn different types of points that allow them to request feedback on their own documents. The recommendations users can obtain from others thus serve as clear objectives for players, with rewards functioning as sources of motivation. Further gamification features as illustrated in Figure 3.5d include a level system and badges that users can earn.

Diabetes self-management In [CCH$^+$12], the authors present a gamified diabetes self-management smartphone application for adolescents called bant (Cafazzo et al., 2012), whose main purpose is to increase the frequency of conducted blood glucose readings provided via a Bluetooth dongle. To that extent, users are not only awarded with virtual points for successive blood glucose testing, but also with rewards with monetary value in the form of smartphone apps and other digital content as illustrated in Figure 3.5e. Using a small sample of adolescents, the authors were able to confirm that the expected increase in the frequency of glucose testing could be observed.

Homework. In [Goe13], the author presents a gamified version of WeBWorK (Mathematical Association of America, 1994), an online homework submission system with a focus on mathematics courses. The system is extended with the integration of two game design elements: experience points and achievements. Specifically, students are rewarded with five experience points for each correct answer, and can advance proficiency levels at certain thresholds whereupon they earn extra credit for their homework score. Furthermore, users may also complete secondary goals such as handing in a homework within 24 hours or submitting a solution with less than five incorrect answers to unlock achievements as shown in Figure 3.5f. The author utilized the implemented system in a Calculus course and found that most students actively engaged with the implemented features and found them motivating and engaging.

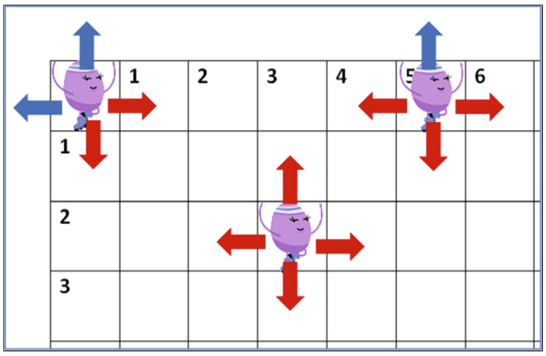
3.3 Related Concepts

Gamification is not an isolated phenomenon, but part of a larger, ongoing transformation that has been coined the *ludification of culture* [Rae06, DDKN11]. Just like the word “gamification”, this term can be decomposed into the Latin word *ludus*, which can be translated to “play” or “game” [Gla68], and the suffix *-ification*. Thus, “ludification of culture” represents a process by which our society is becoming increasingly “ludic”, i. e., playful [Bou12], and through which games and play are becoming an increasingly important cultural phenomenon [Rae06]. Following Huizinga, this change is enabled by the fact that play precedes culture and helps shaping it, and is thus part of the nature of our species, the *homo ludens* [Hui49]. Within the ludification of culture, gamification is just one of multiple directions being pursued by academics and practitioners, albeit currently one of the most popular (cf. Table 3.3). Further fields and concepts that can be related to or differentiated from gamification are described in the following. It should be mentioned that this list is not intended to be exhaustive, as new terms and areas as well as synonyms for existing concepts emerge on an ongoing basis.

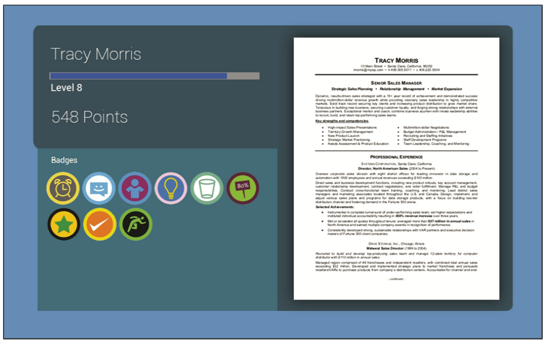


(a) Quick Quiz. Source: [CCF13].

(b) emoticon-bin. Source: [BAZN13].

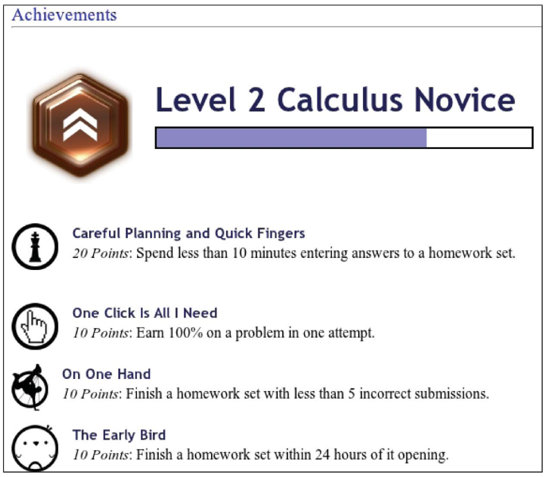


(c) Ariadne PathLogin. Source: [EB16].



(d) Kindle. Source: [vdKK15].

(e) bant. Source: [CCH+12].



(f) Gamified WeBWorK. Source: [Goe13].

Figure 3.5: Screenshots of sample gamification applications from academia.

Table 3.3: List of concepts related to gamification.

Concept	Seminal Publication
Alternate Reality Games	[Szu05]
Digital Game-based Learning	[Pre01]
Games with a Purpose	[vA06]
Gamification	[DDKN11]
Pervasive Games	[MSW09]
Serious Games	[Abt70]

Serious Games

As the name indicates, a serious game is a (digital or non-digital) game whose primary purpose is something other than entertainment [MC05]. The main difference between serious games and gamification is that whereas the former is concerned with *full-fledged* games, the latter is defined around the use of *elements* of games and may not result in a product that is experienced as a game proper [MC05]. The term “serious game” has a longer history than gamification and was first used in 1970 by Abt [Abt70], which is reflected in the higher number of publications in the field. One popular example for this type of artifact is Foldit, an online multiplayer game in which players collaborate and compete in optimizing protein structures (see Figure 3.6), thereby becoming amateur scientists [CKT+10]. Overall, serious games have been used in a variety of different domains and areas, including business, engineering, health, science, and social issues [CBM+12]. With regard to impacts, a meta-analysis conducted by Wouters et al. found training with serious games to be more effective due to the persistence of gains in the long term, but were unable to determine a positive effect on motivation [WvNvOvdS13].

Game with a Purpose

A Game with a Purpose (GWAP) is a particular type of serious game in which the players collaboratively work on a larger task that cannot yet be solved computationally [vAD08]. Such games are used within the context of *human computation*, a discipline that is concerned with “leveraging human abilities [to] solve large-scale computational problems” [vA09, p.418]. Examples of Game with a Purposes (GWAPs) include the ESP Game (von Ahn, 2006)

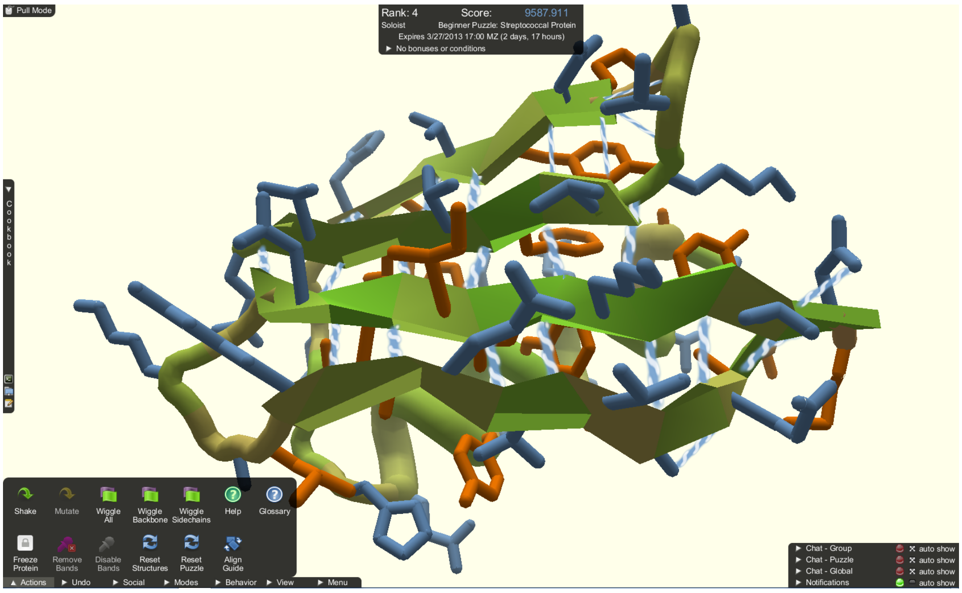


Figure 3.6: Sample screenshot of the serious game Foldit.

(shown in Figure 3.7) and Peekaboom (von Ahn, 2006), two games in which players must analyze and describe the objects they can see in an image for the purpose of tagging [vA06], as well as the game Foldit mentioned previously. GWAPs can be differentiated from gamification in that they also represent full-fledged games and in the specificity of their application.

Digital Game-based Learning

In his seminal book on the potentials of video games for education, Gee states that “you can’t play a game if you cannot learn it” [Gee07, p. 3]. Indeed, many games require a diverse set of skills, such as puzzle-solving capabilities and motor skills. Whereas learning activities in an academic context are often perceived as uninteresting and unenjoyable [RD00a], difficult challenges and the learning processes that allow overcoming them are one of the main sources of fun in video games (see, e. g., [Kos05, SW05, McG11]). Building on this insight, the basic premise of digital game-based learning lies in exploiting the video games industry as “the place where motivation itself is the expertise” [Pre03, p. 1] to keep learners motivated. This is done by integrating commercial off-the-shelf games into learning processes or developing educational games specifically for that purpose [Van06]. In the latter manifestation, digital game-based learning can be seen as a subset of serious gaming.

Figure 3.7: Sample screenshot of the ESP Game.

Pervasive Games

Usually, a game takes place within clearly-defined boundaries that separate it from the real world and in which certain rules creating meaning for the players apply [SZ03]. To play the game, players voluntarily need to cross these boundaries to enter the so-called *magic circle* of the game [Hui49]. One trend within the ludification of culture called *pervasive gaming* sees games extending into the real world, thereby expanding and transcending this magic circle [Mon05, MCMN05]. Such games use information about context of the player (e. g., his location) to provide a gaming experience interwoven with reality and available independently of place and time [BML05]. A recent example of this is Pokémon Go (Niantic, 2016), a game in which players use the cameras of smartphones to capture virtual monsters (called "Pokémon") distributed in the real world. A concept related to pervasive games that blurs the boundaries between the fictional and real worlds even further and is based on collaborative gameplay and engaging narratives are *alternate reality games* [KAL08].

3.4 Game Design Elements

Following the definition of gamification by Deterding et al., one of the core decisions that gamification adopters must make concerns the question *which* game design elements it is exactly that should be used in the application area of interest. As outlined in Section 3.1, these elements constitute the repertoire of gamification that is intended to enable gameful experiences in non-game contexts. Thus, the aim of this section is to provide an overview of the most important elements that game design provides. To that extent, game design elements are analyzed on different levels of abstraction. From most concrete to most abstract, these levels are [DDKN11, p. 12]: *game interface design patterns* (common, successful interaction design components and design solutions), *game design patterns and mechanics* (commonly reoccurring parts of the design of a game that concern gameplay), *game design principles and heuristics* (evaluative guidelines to approach a design problem), *game models* (conceptual models of the components of games or game experience), and *game design methods* (game design-specific practices and processes). Accordingly, the remainder of this section is structured into five subsections 3.4.1–3.4.5 in which each level is first described from a game design perspective. Furthermore, examples are used to illustrate corresponding game design elements in a tangible fashion. Afterwards, this is contrasted with gamification practice as published in a summarized fashion in relevant literature analyses (i. e. [HKS14, MHK16, NZT+14, PGBP15, SA15, TLB14]).

3.4.1 Game Interface Design Patterns

Game interface design patterns are situated on the lowest level of abstraction and relate to the interface of a game, i. e., the part of a game that allows the player to interact with it [Fox05]. In the context of digital games, it is possible to distinguish between the *physical interface* consisting of physical devices such as controllers, keyboards, and monitors used to perceive and interact with the game world, and the *virtual interface* consisting of, e. g., virtual buttons, menus, and progress indicators, which serves the same purpose but resides between the physical interface and the game world [Sch08]. As shown in Figure 3.8, both interfaces allow the player to interact with the game world

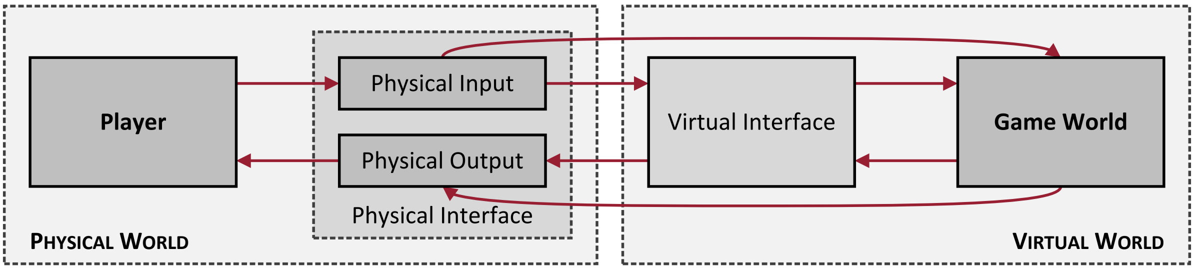


Figure 3.8: Components of game interfaces. Source: based on [Sch08, p. 225].

in a bidirectional fashion. For instance, pushing a thumbstick on a controller may result in a forward-displacement of the in-game character, which will be reflected in the image displayed on the monitor. Similarly, the player may press a button of the virtual interface, thereby casting a magic spell with a temporary effect that is displayed by a small icon in the virtual interface.

Due to its role, the interface must reliably reveal the underlying system of the game to players and allow them to interact with the latter in an intuitive and user-friendly fashion [Jør12]. To that extent, *design patterns* for game interfaces may be very helpful, as they describe recurring problems as well as core solutions to these problems that can be adapted as needed [AIS$^+$77]. Such patterns play an important role in many areas, most notably software design [GHJV95], but of course also game design [17, BH04]. While literature on game interface design patterns is scarce, DETERDING ET AL. refer to the design of social Web sites as a possible source of inspiration [DDKN11]. Indeed, a strong correspondence between patterns presented in relevant literature such as [CM15] and gamification practice can be observed. In particular, three of the game design elements most consistently mentioned across gamification literature (cf. Table 3.4) are interface design patterns: *points*, *badges*, and *leaderboards*. For that reason, these three components are discussed in more detail in the following paragraphs.

Points

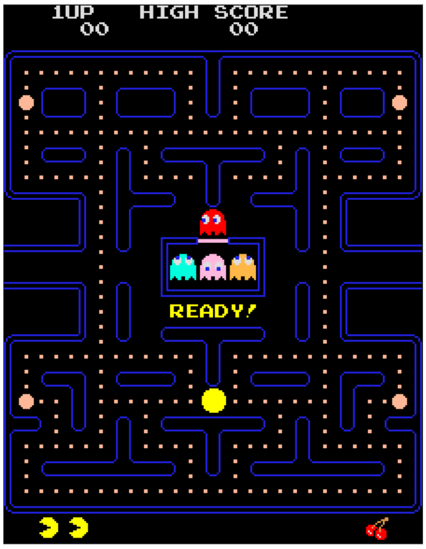
Points are a staple of many games and can be found in a variety of different forms, the most common being that of a *(high) score*. In this manifestation, points primarily measure the success of a player (especially in combination with a ranking system in which they are used to compete with other players), but may have other secondary functions such as unlocking rewards [Sch08].

Table 3.4: Game interface design patterns in gamification practice. Numbers indicate element frequency in literature reviews.

Element	[HKS14]	[MHK16]	[NZT+14]	[PGBP15]	[SA15]	[TLB14]
Points	9	22	✔	15	14	15
Badges	9	10	✔	8	10	28
Leaderboards	10	20	✔	4	12	8

For instance, in the game Pac-Man (Namco, 1980) (see Figure 3.9a), players earn 10 points for every collected small dot, 50 points for every power pellet, further points for other actions, and a one-time bonus life after having reached 10.000 points. As collection is the main purpose of this game, higher scores directly indicate a higher level of player proficiency. Another prominent representative of this game design element that can be found especially in role-playing games are *experience points*. This type of points is typically earned for defeating monsters and completing quests and is used as a measure of character growth. For instance, in Final Fantasy IX (Square, 2000) (see Figure 3.9b), player characters earn experience points for emerging from battles victoriously, which increases character levels at certain experience thresholds. Experience systems are often used in conjunction with *character points*, which can for instance be attribute/status points representing the capability of a character with regard to properties such as physical strength or intelligence. Looking again at Final Fantasy IX, player figures are characterized by four attributes (speed, strength, magic, spirit) which increase over time based on character levels. Further types of points are, e. g., *skill points* denoting the proficiency of a character in special attacks, magical spells, or non-battle skills, *damage points* indicating the strength of a weapon or attack, resources such as *health/hit points* for absorbing damages caused by enemy units, *magic/mana points* for casting spells and using abilities, and *money* for purchasing in-game items, or points related to other game mechanics such as *happiness* and *reputation*.

The use of points in gamification often follows a specific pattern, as many of the examples in Section 3.2 illustrate: First, an overarching goal such learning a language or becoming a better runner is subdivided into smaller activities that are presented to the user. Second, users can then perform these activities, upon



(a) Score in the game Pac-Man.

(b) Experience points in the game Final Fantasy IX.

Figure 3.9: Game interface design pattern: Points.

which they earn points based on some quantitative or qualitative indicator. This may be coupled with a feedback system that provides users with real-time information about their task performance. Finally, points may also serve as a gateway for certain rewards that the gamified system offers, such as ranking highly on a leaderboard or unlocking a badge. Many gamification approaches that practitioners propose are based on these particular steps ([Det15], cf. [ZC11, WH12]). When using point systems in both games and gamification, it is important to consider whether points are actually valuable to players [Sch08]. If they serve no other purpose than providing a quantitative measure for activity, it is very likely that users will not care about points and thus do not exhibit the behavior that the point system is supposed to motivate. In Pac-Man, this meaningfulness is for instance achieved by coupling progression to points collection (i. e., the user cannot advance to the next level without having collected all dots in a level), awarding players for an extra life upon collecting 10.000 points, and ranking players on a high score list according to their points. Analyzing a sample of 24 empirical gamification studies, Hamari et al. find points to have a positive effect on the quality of completed tasks, task completion speed, and the quantity and diversity of user contributions across a variety of different contexts [HKS14]. Furthermore, despite the frequent argumentation that points as an external goal may diminish the intrinsic interest of users in the task being gamified, research suggests that this is not necessarily the case if they are properly used in combination with other game design elements [MBOT13a, MBOT13b].

Badges
Badges—also called achievements, and less frequently trophies, medals, or stickers—are secondary reward systems nowadays common in digital games that provide players with optional goals (e. g., "complete the game", "finish 10 levels without dying", "collect 100 documents") to accomplish in return for virtual rewards that can be displayed to others. As the conditions for unlocking badges often require exploration, excellence, and endurance, this type of rewards can extend the lifetime of a game with comparably low effort [MSW09]. In the context of gaming, badges are often implemented as *achievement systems* on the level of online platforms such as Xbox Live (Microsoft, 2002), PlayStation Network (Sony Interactive Entertainment, 2006), or Steam (Valve Corporation, 2003) rather than the individual game level, although exceptions naturally exist. Following Antin and Churchill, positive effects of badges can manifest themselves in at least five different ways [AC11]:

- **Goal setting**. In this function, badges are challenges for users to overcome. Such goals can be a strong motivator and play an important role in several theories referred to by gamification publications, such goal-setting theory (see Section 3.5). Research indicates that badges are most effective when the goals they represent are achievable and users receive feedback on their progress. Consequently, it has been observed that users exert more effort when they are close to unlocking a badge [MK14].

- **Instruction**. Through their reward conditions, some badges provide users with information about which actions are possible in the gamified system and, more importantly, which actions represent highly-valued behavior. Thus, badges can help with shaping user activity as desired. Furthermore, the ability to view a full list of available badges enables users to gain a holistic understanding of the gamified domain.

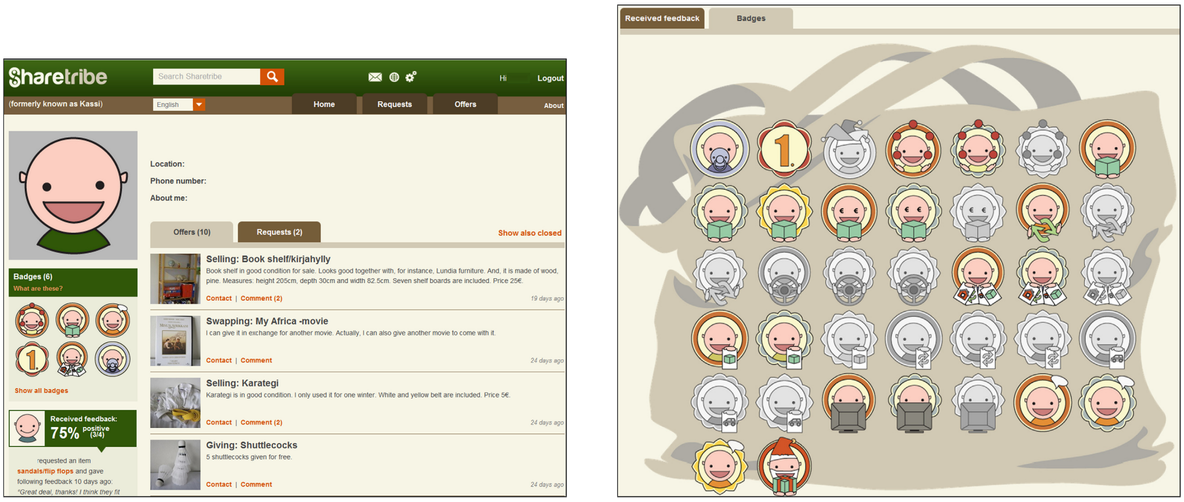
- **Reputation**. Badges can provide information about the interests, skills, expertise, and behavior of the users who have obtained them. Thereby, they allow assessing the reputation of a user, help with assessing the trustworthiness and reliability of any content produced by the user of a gamified system, and can serve as a substitute for direct interaction.

- **Status and affirmation**. Badges can serve as a status symbol that allows users to communicate their accomplishments to others. In this context, the expectation of how a badge is perceived by others is more important than its actual impact. Besides status, badges may also reaffirm users themselves by reminding them of past milestones they have achieved.

- **Group identification**. By means of their reward conditions, badges define subsets of users who have undergone the same trial and thus share certain experiences. Through this, badges may cause a sense of positive identification and solidarity within the group of users who have earned them.

The realization of badge systems in gamification is illustrated in Figure 3.10 by example of *Sharetribe*, a gamified online marketplace with a strong focus on local communities [Ham17]. The users of Sharetribe can earn badges for core activities within the service, such as general use, posting trade proposals, completing transactions, and asking or answering questions. In terms of the classification scheme by Antin and Churchill, most badges are thus associated with goal setting as the major design intention. The badges that a user has unlocked so far can be seen on the user's profile page as depicted in Figure 3.10a. Furthermore, a separate page on which badges that have not been unlocked yet are included is accessible as well and shown in Figure 3.10b. While the badge systems in other examples of gamification may differ in individual details, most adhere to this overall design and present badges to users in a similar fashion.

Due to the popularity of badges as a central gamification component, the question *if* badges work is of central importance. However, the results of empirical studies as summarized in [Ham17, Table 1] paint a mixed picture, often with contradictory outcomes. For instance, while some research found badges to increase user activity, others found this only to be the case when users monitored their badges actively, or not at all. Similarly, whereas some studies observed a positive impact of badges on the quality of the conducted work, other examples were unable to produce these results. Furthermore,



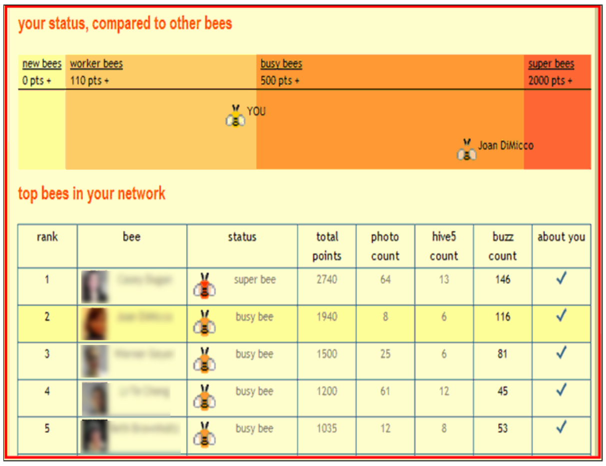
(a) User profile in Sharetribe. Source: [Ham17]. **(b)** List of badges in Sharetribe. Source: [Ham17].

Figure 3.10: Game interface design pattern: Badges.

while statistically significant effects could be observed for some badges, this was not the case for others. Additionally, whereas the prospect of earning achievements seems to be motivating and engaging for some users, others appear to exhibit negative feelings towards them. Clearly, the variation in results can to a certain extent be attributed to differences in implementation. For example, a study with exceedingly negative results conducted by Hanus and Fox in an educational context found badges to be connected to lower motivation, satisfaction, empowerment, and exam scores [HF15]. However, this may be due to shortcomings in the deployed tools which require students to submit badge completion forms and employ a minimalistic spreadsheet for presentation.

In summary, the diversity of results suggests that the focal question should not be *whether* badges work, but rather *how* badges can be optimally designed and aligned with the target users as well as the underlying non-game context. Recommendations to that end include the individual possibility to disable badge systems and coupling badges to interesting, fun, and unexpected reward conditions rather than to how many points a user has earned [HIHK14]. In this connection, research exploring the theoretical foundations of badge systems (e. g., [MK14, Ham17, MK15]) and aiming to define appropriate mathematical models (e. g., [AHKL13]) may be of increasing importance in the future.

(a) Highscore list in Tetris.



(b) Leaderboard in Beehive (IBM, 2008).

Figure 3.11: Game interface design pattern: Leaderboard.

Leaderboards

The underlying idea of a leaderboard—also called ranking or high score list—is to foster competition between the players of a game by using some quantitative indicator of player performance as a means of bringing them into an order from best to worst. Leaderboards are a staple of many digital games, especially those where collecting points is the main goal. An example is the game Tetris (Alexey Pajitnov, 1984), in which players must steer and rotate shapes of various sizes falling down onto the playing field so that the screen fills up as slowly as possible. The main method to score in this game is to fill up horizontal lines, upon which they are removed, thereby prolonging the game and yielding points for the player. The more lines the player manages to delete with a single shape, the larger the size of the reward becomes. Once there is no further vertical space for additional shapes, the game terminates. If the player has managed to earn enough points, he may register himself on a high score list with a name as shown in Figure 3.11a. Thus, the desire to beat other players to earn first place may be a motivator to continue playing Tetris.

In the context of gamification, leaderboards are typically an extension of point systems. For instance, on the corporate social networking site *Beehive*, employees can earn points for uploading pictures, sharing lists, making comments, and adding information about themselves [FDM+08]. Based on the total number of points, users are then ranked on a leaderboard limited to their own personal networks as shown in Figure 3.11b. Leaderboards can drive users in various ways, for instance by motivating them to aim for a top position, not to be worse than others, or not to be seen as inactive users [FDM+08]. This can lead to a higher user activity that is maintained for a longer amount of time when compared to points [MBOT13b]. Furthermore, Landers et al. suggest that users orient themselves towards the top end of a leaderboard and set it as a goal for themselves to overcome [LBC15]. Finally, this game element may be most appropriate for rather simple tasks [LBC15] as the attitude of users toward it is influenced by their perceived self-efficacy in the underlying activity [WKH15].

While the implementation of a leaderboard might seem very straightforward at first—simply rank all users according to their points—some intricate details must be considered. For instance, it has been shown that men are more competitive than women, which indicates that ranking users might not be equally effective for both genders [CG09]. Furthermore, gamification practitioners note that while leaderboards might be motivating for users with a realistic chance to climb the ladder, the opposite effect might occur for those who perceive themselves as incapable of doing so [WH12, KBM14]. Thus, it may be reasonable to consider a larger design space of possible types of rankings, provide more than one leaderboard at the same time, and allow users to opt out of this type of competition. Possible variations of user rankings include leaderboards that are limited to the friends or colleagues of a particular user, leaderboards with temporal restrictions (e. g., only the points earned in the last seven days are considered), leaderboards with limited visibility (e. g., users can only see five users before and after themselves), and aggregated leaderboards for groups of users, such as teams or departments (cf. [KBM14]).

3.4.2 Game Design Patterns and Mechanics

Game design patterns and mechanics are common building blocks of the design of games that enable and give rise to *gameplay*, the “formalized interaction that occurs when players follow the rules of a game and experience its system through play” [SZ03, p. 303]. Compared to game interface design patterns or game design principles and heuristics, elements of this level are thus more specific in the sense that they directly define the set of actions that players can perform in a game. Consequently, there can be strong variations between different games and games from different genres. For instance, the mechanic “jumping” is essential for nearly all platform games, but cannot be found in most role-playing games, although exceptions may always exist (e. g., Castlevania: Symphony of the Night (Konami, 1997), a side-scrolling game combining elements of both, platform games and role-playing games). The following paragraphs provide more detailed information about game mechanics and game design patterns.

Game Mechanics

In the context of games, mechanics define the actions and behaviors that players may perform within the context of a game, thus providing a specification of its rules [HLZ04]. Therefore, as Schell puts it, they represent the “core of what a game truly is” [Sch08, p. 130] beyond its aesthetics, technology, and game contents such as artworks or story. This shall be illustrated by example of Super Mario World (Nintendo, 1990) (see Figure 3.12). In this game, players must restore peace to Dinosaur Land and rescue Princess Peach from the evil Bowser by navigating the player character, Mario, through up to 74 levels from left to right. The basic mechanics of the game consist of *walking*, *running*, and *jumping*, which the player must combine to reach the end of each *level* while overcoming various *obstacles*. In this process, players may encounter a variety of *enemies* with different movement patterns. In most cases, jumping onto an enemy will result in its *defeat*, whereas colliding with the enemy will result in the *death* of the player character. Upon death, the player loses one of initially five *lives*, the loss of all of which will result in a *game over*. Players have to finish each level within a given *time limit*; if they fail to do so, they lose a life as well. A third way for the player to lose a life is to let Mario *fall* into a chasm or a pit filled with lava. Jumping against certain blocks from below may reward players with a *coin* or reveal a *power-up* for Mario. The latter equip Mario with capabilities such as shooting enemies with *fireballs*, being *invisible* for a while, or *flying* for a short amount of time. Furthermore, for every 100 collected coins the player earns one additional life. Levels are subdivided into seven *worlds*, of which each has a special *boss* to defeat at the end. During his quest, Mario may sometimes receive help from Yoshi, a dinosaur on which the former can *ride*. Finally, many actions in the game yield *points* which eventually add up to

Figure 3.12: Screenshot of the game Super Mario World.

the high score of the player. This is only a selection of the mechanics of Super Mario World, and numerous further intricacies can be found in the game.

Generally speaking, creating an exhaustive taxonomy of game mechanics is not possible as (even simple) games are very complex systems whose individual components interact in many ways, thereby making them difficult to unravel [Sch08]. However, even an incomplete list of mechanics might be useful for game designers, and thus examples can be found in many relevant books. For instance, Schell proposes a subdivision of game mechanics into the categories *space*, *objects*, *attributes*, *states*, *actions*, *rules*, *skill*, and *chance*. Adams and Dormans instead distinguish between *physics*, *internal economy*, *progression mechanisms*, *tactical maneuvering*, and *social interaction* [AD12]. Further examples can be found, e. g., in [Bat04], [SZ03], and [Ful08].

Focusing on the context of gamification, Deterding et al. mention time constraints, limited resources, and turns as some examples for game design elements on this level [DDKN11]. While game mechanics can indeed be found in some gamified solutions (for instance, Duolingo includes both timed lessons and limited lives), none of the literature reviews mentioned at the beginning of this chapter names game mechanics among the most frequent elements. This may be due to the fact that the non-game context itself often imposes an initial set of mechanics on designers, thereby prompting their exclusion from what is considered to be the design space of the solution. For instance, while “running” can generally be seen as a game mechanic of Super Mario World (Nintendo, 1990), it might not be perceived as such for Nike+. Consequently, the lack of explicit discussion regarding game mechanics in gamification literature may be a problem of reporting rather than usage.

Game Design Patterns
Design patterns represent a more problem-oriented approach to game design in which typical design problems are described together with accepted solutions, which may in turn be mechanics. Kreimeier proposes that the description of a game design pattern should (at least) include its name, the problem to solve, the solution, and its consequences [17]. An example for such a pattern, namely *paper-rock-scissors*, is presented in Figure 3.13. Here, the basic idea lies in ensuring that there is no single strategy for players that is optimal in all situations. Further game design patterns can be found in [BH04] and [AD12], although literature on this topic is more scarce in general. In addition, Adams and Dormans propose *Machinations* as a domain-specific process modeling language for game mechanics and design patterns with a formal syntax that allows simulating the underlying models [AD12]. Machinations have also been discussed in the context of gamification by Herger, who suggests that gamifiers can use them to test and balance their gamification designs [Her14]. Beyond this, little discussion on game design patterns is offered by gamification literature to date.

3.4.3 Game Design Principles and Heuristics

Game design principles and heuristics consist of intuitive guidelines that can be used to address specific game design problems. In contrast to design patterns, *design principles* only provide general guidance by encapsulating knowledge about best practices, but do not prescribe particular solutions [CM15]. Similarly, *heuristics* aim to leverage intuition as a source of knowledge for making reasonable decisions in the light of complex problems to achieve certain goals [Pea84]. An example for such a guideline is “provide immediate feedback for user actions” [DCT04, p. 1511]. One potential source for game design elements of this level are general books on game design, such as [SZ03],

PAPER-ROCK-SCISSORS

Problem: Avoid a dominant strategy that makes player decisions a trivial choice.

Solution: Introduce nontransitive relationships within a set of alternatives, as in the game of paper-rock-scissors.

Consequence: The player is no longer able to find a single strategy that will be optimal in all situations and under all circumstances. She has to revisit her decisions, and, depending on the constraints imposed by the game, adjust to changing situations, or suffer the consequences of an earlier decision.

Examples: The example given by Andrew Rollings is the set of warrior-barbarian-archer from the Dave and Barry Murray game *The Ancient Art of War* (Broderbund 1984). He also describes *Quake*'s weapon/monster relations in similar terms: Nailgun beats shambler, shambler beats rocket launcher, rocket launcher beats zombie, zombie beats nailgun [28].

References: Chris Crawford (see "Triangularity" in [15]) provided the first explicit description of the use of nontransitive relationships. Andrew Rollings' discussion of examples uses game theory including detailed payoff, as well as informal fictional designer dialogs.

Figure 3.13: Game design pattern "paper-rock-scissors". Source: [17].

[Ful08], and [Sch08], even though they might only discuss principles and heuristics implicitly. Furthermore, a range of academic works has proposed sets of heuristics aimed to guide game development, with some focusing on game design in general [Fed02, DW09], whereas other authors have focused on heuristics for more specific topics, such as the design of instructional games [Mal80], the design of games that afford flow experiences [SW05], or the playability of games [DCT04, PWS08]. From these sources, the following lists of game design principles and heuristics is compiled:

- **Enduring play**. A game should be fun and provide long and enduring gameplay as well as replayability. It should not contain uninteresting, repetitive, or unimportant tasks. Activities and pacing should be varied to reduce boredom and fatigue. Frustration should be reduced by not penalizing players for the same mistakes over and over again, e. g., by letting them skip over non-interactive content such as cutscenes.
- **Challenge**. A game should have an uncertain outcome; it should neither be so hard that players know they will fail, nor so easy that success is

guaranteed. Hence, the level of challenge should match the skills of the players to put them under pressure without causing frustration. The level of challenge should rise over time with increasing player skills and mastery. To accommodate players of varying skill levels, the difficulty level should adapt automatically, or players should be able to change it.

- **Skill acquisition**. A game should be easy to learn, but hard to master. Skills required to overcome challenges should be taught to the player early enough or directly before they are needed. Learning these skills should not be tedious and uninteresting, but part of the fun.

- **Clear goals**. A game should have clear goals with multiple levels of granularity and complexity. A clear, overriding game goal should be presented to players early on. Smaller, more short-term goals should be introduced at appropriate points in the game.

- **Feedback**. Players should receive feedback that illustrates their score and status in the game as well as their progress towards achieving the goals of the latter. This feedback should not interfere with gameplay and provide players with all information they need to make good decisions while continuing to play. Furthermore, feedback should be given immediately upon player actions to heighten the player's sense of control. Lastly, it should be noted that feedback can be multimodal, i. e., visual, visceral, and aural (music and sound effects).

- **Rewards**. Players should receive appropriate rewards for the effort they invest into the game. One type of reward should be the acquisition of certain skills while playing. Other rewards should increase the immersion of players in the game by expanding their set of in-game actions, for instance through new capabilities or customizations.

- **Beginning of play**. It should be possible to start playing a game without reading a manual or documentation. There should be a tutorial or simple initial levels that teach players how to play, are interesting and absorbing, and feel like playing the actual game. However, players should be able to skip this initial content if they wish to. The beginning of the game should be very obvious and easy to play, thereby resulting in positive feedback independently of the skill level of the player.

- **Play styles**. A game should offer multiple ways to win, present multiple paths to the goal, and cater to multiple different styles of playing. Different ways of playing should be balanced so that there is no single dominant strategy.

- **Player support**. To cope with potentially complex mechanics and steep learning curves, games should provide some means of support to players. For instance, players should be able to receive hints that help with overcoming challenges. New game mechanics should be taught to players through interactive tutorials. Context-sensitive help should provide guidance to players when and where they need it. Finally, it should be possible to access the documentation of the game without exiting it.

- **Player control**. Players should feel in control of their characters or units, their movements and interactions, the actions they take, the strategies they use, and the user interface and input devices. Furthermore, they should have the sense that they have an impact on the game world. Players should be prevented from making errors that gravely affect gameplay and should be supported with error recovery. They should be able to save the game in different states so that play can be easily interrupted and resumed.

- **Controls**. The controls of the game should be easy to learn and intuitive. Furthermore, they should follow standards already established in the game industry and in certain game genres. Advanced players should be able to customize controls according to their needs. Responses to player inputs should be appropriately sensitive and responsive and lead to consistent and expected outcomes.

- **Interface**. The user interface should be non-intrusive and, if possible, hidden during gameplay. Otherwise, it should be experienced as part of the game as much as possible. The layout of the interface should be efficient, pleasing, and consistent in design and control. The number of menu layers should be minimal and navigation intuitive.

An example that illustrates how successful games incorporate these heuristics can be found in Diablo III (Blizzard Entertainment, 2012), a critically acclaimed Action RPG with an average rating of 88% on the review aggregator Web site *metacritic*[2]. The main activity in this game performed by the player lies in navigating a virtual character through a fantasy world while defeating monsters using a wide arsenal of weapons and skills. The level of challenge increases over time as enemies gain in strength and more demanding difficulty levels become available. Analogously, the player character becomes more capable as well by gaining levels, finding new weapons and armor, and unlocking new skills. As seen in Figure 3.14, Diablo III provides feedback to players in many different ways: small red bars representing enemy health and numbers displaying player damage (box A), interface elements indicating, e. g., player health, active skills adding certain effects, and progress towards the next character level (box D). Further feedback is provided through visual effects and sound effects. The game is organized in terms of quests that guide the player throughout the story of Diablo III. Each quest provides the player with clear goals, such as navigating to a particular location, finding an item, or defeating a boss monster (boxes B and C). Monsters eliminated by the player may randomly drop various kinds of rewards, most notably so-called "legendary" items that are very rare and highly effective. Lastly, Diablo III supports a wide variety of play styles by offering different classes (e. g., barbarian, wizard, and monk) that can be played in different configurations (so-called builds).

Already long before the advent of gamification, Malone has examined how the design of enjoyable software user interfaces can be informed by game design [Mal82]. He suggests that applications should present users with clear goals, provide performance feedback, have a variable difficulty level, and stimulate the users' fantasy and curiosity—all of which can be connected to the heuristics listed above. Extending these insights to gamification is a natural consequence, and thus it is hardly surprising that game design elements of this level can frequently be found in academic publications on gamification as seen in Table 3.5. On the basis of this data, the main idea pursued by gamifiers can be summarized as follows: the users of a gamified system should be provided with challenging tasks with clear goals, should receive feedback on their progress while working on them, and successful completion should yield

[2] See: http://www.metacritic.com/game/pc/diablo-iii. Last accessed: 2016-12-01.

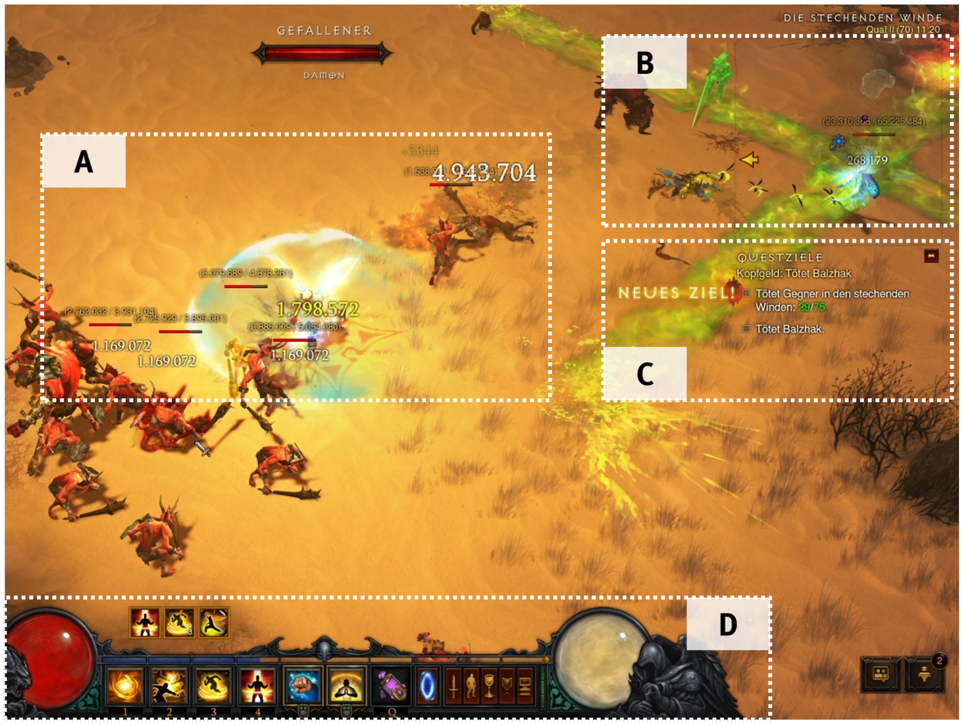


Figure 3.14: Screenshot of the game Diablo 3.

rewards, which may in turn be connected to interface elements such as points and badges. This is also reflected in books by gamification practitioners, which make frequent use of terms such as challenge, feedback, goals, and rewards (cf. [WH12, Bur14, Her14, KBM14]). Further examples for game design elements on this level mentioned by Deterding et al. but not reflected in the literature reviews include enduring play and enabling a variety of game styles [DDKN11]. Lastly, some authors have also proposed domain-specific sets of gamification design principles, for instance for enterprise gamification [OJK14] and the gamification of health [PTC13].

3.4.4 Game Models

According to Deterding, game design elements of this level relate to conceptual models that describe the individual components of which games are comprised and that characterize the experience of playing a game [DDKN11]. Besides academic publications, such models can also be found in many books on game design, for instance [SZ03], [Ful08], [Sch08], and [Rou05]. Generally

Table 3.5: Game design principles and heuristics in gamification practice. Numbers indicate element frequency in literature reviews.

Element	[HKS14]	[MHK16]	[NZT+14]	[PGBP15]	[TLB14]
Clear goals	4	2	✖	1	23
Feedback	6	4	✔	✖	12+4
Rewards	4	3	✔	✖	6
Progress	4	5	✔	✖	6
Challenge	7	✖	✖	✖	✖

speaking, game design elements on this level are often subjective, highly interrelated and overlapping, and in many cases not validated empirically. In the following three subsections, three (types of) models that are often discussed in foundational gamification publications are presented.

Game Components

To create gameful experiences, it seems prudent to first gain an understanding of what a game actually is. In literature on game design, many different, albeit often overlapping definitions of the term "game" can be found. For instance, a rather simple characterization is given by Schell: "A game is a problem-solving activity, approached with a playful attitude" [Sch08, p. 37]. Furthermore, the author states that games consist of four types of elements, namely mechanics, story, aesthetics, and technology. Another definition is provided by Salen and Zimmerman: "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome" [SZ03, p. 96]. McGonigal does not propose an explicit definition, but identifies four characteristics of games: goals, rules, feedback systems, and voluntary participation [McG11]. She then goes on to describe the act of playing a game as "the voluntary attempt to overcome unnecessary obstacles" [McG11, p. 22]. Fullerton considers a game to be a "closed, formal system that engages players in structured conflict and resolves its uncertainty in an unequal outcome" [Ful08, p. 43]. By "closed", the author states that games are separated from the real world, in which they have no consequences. Lastly, synthesizing and extending the definitions of many other authors, Juul proposes the following definition: "A game is a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable" [Juu11, p. 36]. In comparing these definitions to the intent and context of gamification, two major differences can be identified: Firstly, whereas games are played voluntarily, the use of a gamified system can also be mandated, for instance in an organizational setting. Secondly, while games take place in a space separated from reality, real-world consequences are one of the defining characteristics of gamification.

Designing gameful experiences requires an understanding of which components games consist of and how the experience of playing games works. In this context, the Mechanics, Dynamics, Aesthetics (MDA) framework may be a helpful tool. The MDA framework was designed as a formal approach for explaining and understanding games using a shared vocabulary for game designers, critics, and developers [HLZ04]. The framework distinguishes three essential game components, namely *mechanics* which are the individual components of a game that define the sets of actions and behaviors that players may perform and exhibit (see Section 3.4.2), *dynamics* that correspond to the run-time behavior of mechanics as they are put into action through player inputs and outputs, and lastly *aesthetics*, which represent the desired emotional responses that should be evoked from players. The framework further distinguishes between two essential roles: the game *designer* who is responsible for crafting the game, and the *player* for whom it is created and who eventually consumes it. Both of these actors approach the game from a different perspective: whereas the designer can only influence its mechanics, and the dynamics and aesthetics naturally unfold from the former, the player first experiences aesthetics which result from dynamics and mechanics in motion. In gamification literature, the MDA framework is often employed as a means for structuring the discussion on game design elements (e. g., [ZC11, TLB14, BVW15]). However, Deterding suggests that its consideration of the "emergent, systemic quality of game enjoyment" [Det15, p. 300] makes it more appropriate as a guiding principle for the iterative design of gamified solutions using mechanics to evoke certain aesthetic experiences [Det14, Det15]. Lastly, it should also be noted that authors often misunderstand the MDA framework, for instance by misclassifying certain elements as dynamics rather than mechanics [7, Det15].

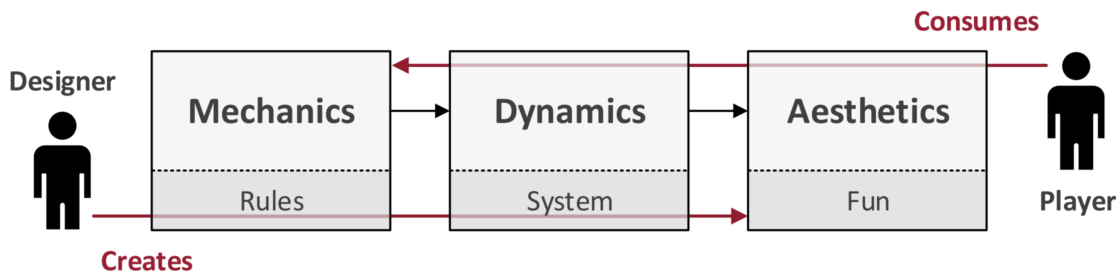


Figure 3.15: Mechanics, Dynamics, Aesthetics framework. Source: based on [HLZ04].

Enjoyment

As Przybylski et al. aptly put it, "the appeal of video games lies in the inherent properties of the experiences they provide" [PRR10, p. 155]. Consequently, studying the different ways in which games entail engaging and enjoyable experiences as well as the particular properties of games that result in the former is of considerable importance for game design and development. Naturally, this also holds for gamification, as its underlying idea is to recreate such gameful experiences in other contexts. Some of the different proposals discussing types and sources of enjoyment in games are as follows:

- **8 kinds of fun**. As part of the MDA framework, Hunicke et al. also provide a (non-exhaustive) list of eight different aesthetics, i. e., possible ways in which games can be fun [HLZ04, p. 2]: sensation (games as sense-pleasure), fantasy (make-believe), narrative (drama), challenge (obstacle course), fellowship (social framework), discovery (uncharted territory), expression (self-discovery), and submission (pastime). Clearly, these are not mutually exclusive, and games can aim to be fun in multiple ways at the same time. Using this taxonomy to analyze existing game can help with identifying the mechanics and dynamics that enable certain kinds of fun.

- **Core Elements of the Gaming Experience (CEGE)**. Based on qualitative data from video game reviews and interviews, the Core Elements of the Gaming Experience (CEGE) model proposed by Calvillo-Gámez et al. aims to define *hygienic factors* for positive gaming experiences, i. e., factors that are necessary, but not sufficient for game enjoyment

[CGCC10]. Specifically, the authors distinguish between factors relating to the *game* itself, and factors related to *puppetry*, the interaction of the player with the former. As relevant aspects of a game, *gameplay* (rules, scenario, etc.) and *environment* (physical presentation of the game to players, for instance through graphics and sounds) are identified. Furthermore, puppetry is defined via the three conditions *control*, *ownership*, and *facilitators* (i. e., time, aesthetic values, and previous experiences available to players). The CEGE model posits that if all of these factors are present, the experience of playing a game will at least not be negative, even though this does not mean in turn that it will be positive.

- **Four keys to more emotion**. Based on observations of gamers while playing and interviews with non-gamers, Lazzaro proposes that games engage players in four different ways [18]: hard fun, easy fun, altered states, and the people factor. *Hard fun* is related to pursuing goals, overcoming difficult challenges, developing and applying different strategies, and receiving progress feedback. *Easy fun* instead lies in the intrinsic enjoyment of game activities as they pique the curiosity of players and entice them to find out more about the game world through awe, mystery, and wonder. *Altered states* refers to how games can function as a therapeutic device by changing the mental states of players and evoking various emotions in them, including excitement and relief. Lastly, the *people factor* relates to enjoyment that arises when playing with others, both while competing and cooperating. Lastly, Lazzaro notes that people playing in groups tend to express their emotions much more strongly than those playing on their own.

- **GameFlow**. Based on Csikszentmihalyi's research into flow experiences, Sweetser and Wyeth propose the GameFlow model as a means for evaluating player enjoyment in games [SW05]. To that extent, the model conceptualizes enjoyment via the following eight elements [SW05, p. 5f]: *concentration* ("[games] should require concentration"), *challenge* ("[games] should be sufficiently challenging and match the player's skill level"), *player skills* ("support player skill development and mastery"), *control* ("[players] should feel a sense of control over their actions"), *clear goals* ("provide the player with clear goals at appropriate times"), *feedback* (“appropriate feedback at appropriate times”), *immersion* (“deep but effortless involvement”), and *social interaction* (“support and create opportunities for social interaction”). Clearly, the GameFlow model encapsulates a considerable portion of the game design heuristics presented in the previous section.

- **Psychogenic needs**. Based on a previous conceptualization of human needs, Bostan argues that the motivation from playing games arises as they enable players to satisfy certain psychological needs, thereby prompting goal-directed behavior [Bos09]. Specifically, the author mentions *materialistic needs* (acquisition, construction, order, and retention of objects), *power needs* (aggression, counteraction, defendance, dominance, etc.), *affiliation needs* (affiliation, rejection, nurturance, etc.), *achievement needs* (achievement, autonomy, recognition, etc.), *information needs* (cognizance, exposition, understanding), and *sensual needs* (play, sentience, sex) as sources of motivation.

- **Player Experience of Need Satisfaction (PENS)**. Similarly to the work of Bostan, the Player Experience of Need Satisfaction (PENS) model also argues that games are motivating due to their capability to satisfy human needs and the human tendency to seek out activities that satisfy the latter [RRP06]. However, Ryan et al. instead assume the perspective of Self-determination Theory (SDT), which argues that the three universal, innate psychological needs are *autonomy* (acting willingly and out of free choice), *competence* (being challenged and overcoming challenges), and *relatedness* (feeling connected to others). The authors extend this theory with two additional factors: *presence* (the sense of being embedded within a game world) and *intuitive controls.*

For additional information on enjoyment in games, the reader is referred to the studies conducted in [BCHB12] and [HK17]. By considering models such as these, gamification designers can aim to create solutions that afford enjoyable experiences similar to those presented by games. Deterding suggests that in particular the PENS model is of particular importance, as its foundation is formed by SDT, a “well-established theory of human motivation with empirical support across contexts” [Det15, p. 298]. Furthermore, he argues that PENS is general enough to be able to also explain the constructs of other models. Looking at gamification practice as reported in relevant literature reviews, there are strong indications that academics prefer directly employing the theories underlying some of the models mentioned above (e. g., Flow theory as the basis of GameFlow, or SDT underlying the PENS model) rather than the models themselves [SA14, PT15, SF15b]. Thus, the most important theories used to explain the impacts of gamification are discussed separately in Section 3.5.

Player Types

When developing a game, the game designer always acts as the advocate for the player, and thus his main task is to define goals for the types of aesthetics (in terms of the MDA framework) that players will experience while playing the game [Ful08]. Any subsequent discussion of game features should then be conducted with these goals in mind. However, it must be acknowledged that different players might play games for different reasons, and thus what may be fun for some, may in turn constitute a frustrating experience for others. One influential work in the area of player segmentation is the framework of Multi-User Dungeon (MUD) player types by Bartle, who, as Figure 3.16 depicts, distinguishes between *achievers* who mainly wish to act on the game world to reach mastery, *explorers* who wish to interact with the game world to discover its intricacies, *socializers* who want to interact with other players in an amicable fashion, and *killers* who are interested in demonstrating their superiority over other players [Bar96]. This model is discussed in many gamification textbooks [WH12, Her14], and sometimes used to propose mappings of game design elements to classes of players for whom they might be appropriate [ZC11, Mar15]. However, as Deterding notes, this is problematic, as there is no empirical evidence for its usefulness beyond the particular context of MUDs [7]. Thus, a suggested alternative lies in the use of contextually-situated, data-driven, and research-based personas [Dix11, Det15]. In interaction design, the term *persona* denotes an “archetype of a user that is given a name and a face, and it is carefully described in terms of needs, goals and tasks” [BA02, p. 197]. Consequently, the task of the gamification designer lies in conceptualizing a solution that satisfies the needs and goals of the persona. For more detailed information on player typologies, the reader is referred to [Dix11] and [HT14].

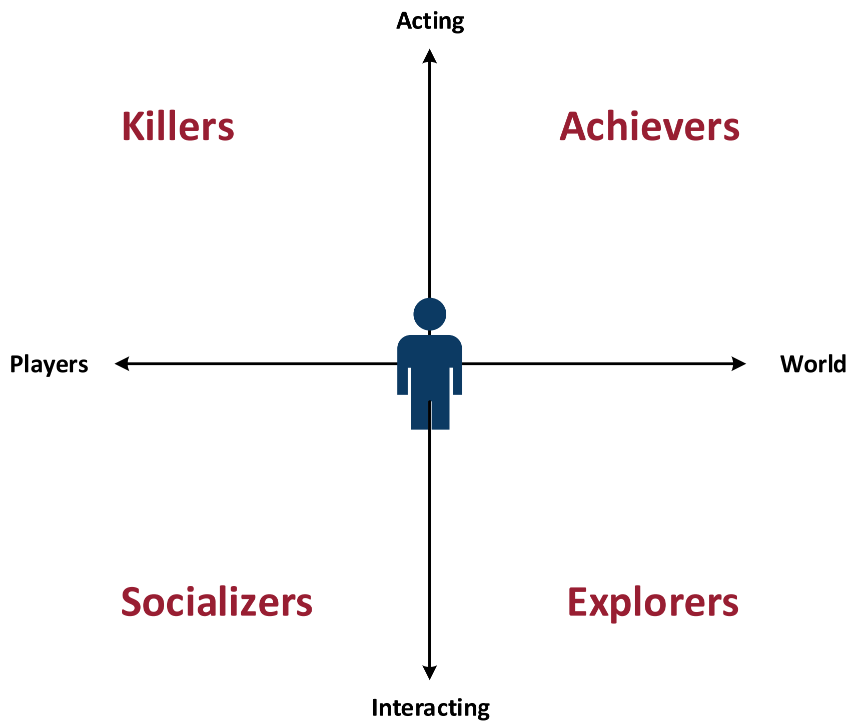


Figure 3.16: Bartle's Multi-User Dungeon player types. Source: based on [Bar96].

3.4.5 Game Design Methods

Game design elements on this level consist of practices and methods that are specific to the game design discipline [DDKN11]. Just as any other corporate activity, game design is a business process consisting of various, interrelated activities. Such processes are discussed in many books on game design, albeit mostly in textual form and often together with more technical development activities. Synthesizing the processes described in various game design books (i. e., [Bat04, Ful08, Rou05, SZ03, Sch08]), Dzgoeva derives the following list of top-level activities, each of which can be further detailed on the sub-process level [Dzg16, p. 24-31]:

1. **Develop concept**. First, a game idea is generated, which may be based on technology, story, or gameplay as a starting point. Furthermore, the goals and constraints of the game are defined. Afterwards, the idea is evaluated against parameters such as technical feasibility and budget,

which may result in a termination of the development project. Otherwise, the features and mechanics of the game to be created are designed, experience goals for players specified, and a simple prototype is created to test the former and help with making design decisions.

2. **Execute preproduction**. Once the concept of the game has been approved, digital prototypes of the game are successively created and evaluated until satisfactory results are achieved. This is followed by the creation of various types of documentation, including a design document, a story bible, an art bible, and a technical design document. The completion of this documentation also terminates the preproduction stage.

3. **Execute production**. In the main development phase, the content of the game is created, including its code, arts and animations, dialog, and other in-game artifacts. This is accompanied by project management to ensure that the endeavor remains on schedule and within budget. Once a candidate version of the game has been finished (a so-called alpha version), light testing is performed. Should the quality of the product be found to be unsatisfactory at this point, a new iteration of the production stage is started, which may result in a refinement of the aforementioned artifacts.

4. **Execute quality assurance**. Once the alpha version has been approved, the main testing phase commences. At the core of this activity sits the generation of a test plan and the execution of different types of testing, including focus groups, playtesting, quality assessment, and usability testing. This results in a prioritized list of bugs, which are then resolved in the order of their importance. The result of this stage is the "gold code" of the game, which represents its final version.

5. **Launch game and provide maintenance**. After quality assurance, the game is released in its current state. As it might still contain bugs at this point, player feedback is carefully monitored, and support provided as needed. This may result in the creation of patches, which are published to fix any remaining problems. Furthermore, upgrades for the game may be developed to extend its contents or underlying mechanics.

These activities are typically not carried out in a sequential fashion, but in a repeating cycle of conceptualization, realization, and testing, until the predefined goals of the game development endeavor are met [Det15]. This iterative nature of creating games is also acknowledged in many game design books aimed at professionals [Bat04, SZ03, Ful08, Rou05]. Newell further stresses that “the iteration of hypothesis, changes, and measurement [. . .] will make [your game] better at a faster rate than anything else we have seen” [40]. Many authors emphasize the importance of prototyping and playtesting as crucial tools that should be used from as early on as possible and throughout the entire development project [SZ03, Ful08, Sch08]:

- **Prototyping**. The basic idea of *prototyping* lies in the creation of a functional model of an idea to test its feasibility and make improvements as required. Such a prototype can be digital, but may also have a physical form, i. e., be created using pen, paper, and other physical materials. Prototypes can be compared to rudimentary sketches of a drawing and thus dispense with detailed resources and perfect realizations of features. They can be used as a basis for playtesting and facilitate gaining feedback early on in a development project. Furthermore, prototypes may focus on a particular aspect of a game (e. g., a particular game mechanic, visual style, control scheme, or user interface) that shall be examined in detail.
- **Playtesting**. As game designers are the advocates of players, it is advisable to establish a close relationship between these two groups as early as possible. Otherwise, it may happen that the final product does not properly meet the needs of players and the defined player experience goals. To prevent this, iterative playtesting should be conducted. This means that the designed and implemented concepts are tested with the eventual players throughout the entire development project so that the design of a game can be adapted on the basis of their feedback. Clearly, the largest opportunities for change can be found in the early project phases, whereas the design should converge to a stable state throughout implementation. It should be noted that playtesting is a rather formal activity that goes beyond “just letting people play” and uses a variety of different tools, settings, and participants.

When designing a gamified application, different approaches with varying degrees of sophistication can be chosen, the arguably most simple of which is adding PBL to the non-game context of choice. Another possibility lies in choosing one of the various theories of human motivation whose applicability to gamification has already been demonstrated and trying to create a design concept tailored to the former. For instance, a designer might aim to enable flow experiences by conceptualizing a solution around the GameFlow model discussed in the previous subsection. Looking at practice-oriented recommendations for how to conduct gamification, Deterding identifies the following process by summarizing the proposals of various authors [Det15, p. 306f]:

1. “Identify system owner goals.
2. Identify trackable behaviors of end users that support these goals; quantify their relative contribution in a metric.
3. Profile and segment end users using player typologies (usually “Bartle Types”).
4. Select and specify game design patterns:
    a) Translate the quantified system owner value of end user behaviors into point values displayed back to users.
    b) Articulate an ordered sequence of explicit goals for end users, consisting of sets of behaviors or point thresholds (quests, challenges, levels).
    c) Define feedback to display upon single user actions (“engagement loop”) as well as reaching point thresholds or goals (achievements, badges, leader boards), including potential rewards (virtual items, customization options).
    d) Choose additional game design patterns.
5. Playtest.
6. Build and deploy.
7. Use analytics of user behaviors to monitor system performance and guide the improvement and release of new content and features.”

The author further notes that such approaches are problematic for a variety of reasons [Det15, p. 307-310]: first, they are not sufficiently data-driven, i. e., they do not incorporate measurements and observations of real users and other contextual factors. Second, they rely on player typologies such as those by Bartle that are outdated, not sufficiently validated, and not necessarily applicable outside the context for which they were originally proposed. Third, they too often suggest adding the same elements (most notably PBL) regardless of context. Fourth, they do not provide enough guidance in the choice of (different) game design elements. Fifth, they appeal to motivational psychology, which is often misunderstood and extended in an unsound fashion. Sixth, they do not consider iterative prototyping.

To overcome these problems, academics have proposed a variety of tools and methods for gamification design, such as *skill atoms* [Det13, Det15] and *Gamicards* [FWG14]. Others have suggested that conducting gamification can be seen as a form of design science, and thus methods and guidelines from the latter could also be extended to gamification design [KTCK12]. The most comprehensive gamification process to date was proposed by Morschheuser et al. and already discussed in the introduction of this thesis [MWHA17]. Based on a review of relevant literature and interviews, the authors not only provide a detailed process model (see Figure 1.4), but also list general requirements for gamification projects and possible tools that can be used in the ideation phase to generate gamification designs. For additional information about approaches for gamification design, the reader is referred to a recent literature review presented in [MRGAM15]. Despite the increasing availability and maturity of gamification design methods, it must be noted that the fraction of publications that actually uses them is still negligible, and thus not enough practical experience exists. Instead, most implementers opt for a pattern-based approach and do not explicitly document their design process.

3.5 Theoretical Foundations

Gamification is grounded in the assumption that games have salient properties that are causally linked with their positive impacts, and that transferring these properties to non-game contexts will also carry the benefits over to the latter. However, the mechanism through which this is expected to work is often left unclear, which causes gamification to become a black box with uncertain outcomes. This manifests itself, e. g., in the phenomenon that different implementations of the same basic gamification approach have yielded not only positive results, but have also produced mixed and highly negative outcomes in different empirical studies (see, e. g., [Ham17], [HIHK14], and [HF15] for studies on the use of badges with positive, mixed, and negative outcomes, respectively). Therefore, to gain a better understanding of *how* and *why* gamification works—and thus create a basis for enabling designers to design and develop effective gamified applications—the theoretical foundations of gamification must be examined. To date, the discussion of theory in gamification is still mostly confined to conceptual work, whereas applied research often forgoes proper empirical validation using established psychometric measures [SF15b]. Based on a synthesis of three recent reviews that have analyzed the use of theory in gamification research [SA14, PT15, SF15b], the purpose of this section is to provide an overview of the most important theories that have been accepted into the gamification canon and that are most relevant for its application in Part II.

3.5.1 Intrinsic and Extrinsic Motivation

Research on motivation is concerned with the question of how motives affect human behavior. Someone who is motivated is moved to perform a certain action, and feels activated to intentionally achieve a certain outcome [RD00a]. The nature of motivation may differ with regard to its *level* (strength) and *orientation* (type). The latter is concerned with the different factors that move people to do something, which may either be internal or external [RD00b]. The two main types of motivation that can be distinguished are *intrinsic motivation* and *extrinsic motivation*. Whereas the former denotes doing something because the activity itself is interesting and enjoyable, the latter means to act due to reasons that are clearly separable from the activity itself [RD00a]. For instance, with the exception of professionals, most people perform sports for purely intrinsic reasons [FR95]. In contrast, many of the activities that teachers want students to carry out are neither interesting nor enjoyable to them, and are thus often performed for external reasons such as praise or achieving good grades [RD00a].

Intrinsic motivation reflects the inherent tendency of humans to learn, improve their skills and capabilities, overcome challenges, and experience new stimuli [RD00b]. As such, it is often considered to be a superior kind of motivation [RD00a]. Extrinsic motivators in turn can for instance be monetary rewards, awards, tokens, toys, foods, and prizes, or the avoidance of punishment [DR85]. In early research on motivation (especially work motivation), no distinction between intrinsic and extrinsic motivation was commonly made, or the two considered additive factors of a single motivation construct [GD05]. However, in a meta-analysis of 128 studies, Deci, Ryan, and Koestner have found strongly consistent evidence that extrinsic motivators such as tangible rewards generally undermine intrinsic motivation for high-interest tasks [DKR99]. Some exceptions exist, for instance for verbal rewards, rewards that are neither expected nor dependent on task performance, or for rewards given for low-interest tasks [CP94]. Another aspect to consider is the so-called *overjustification effect*, which posits that once a shift from intrinsic to extrinsic motivation has occurred, a removal of the extrinsic factor will not cause intrinsic motivation to return. Thus, for the desired behavior to continue, the supply of extrinsic motivators may also not subside [LGN73].

Creating motivating and engaging experiences is the central challenge of the video gaming industry, and thus the latter has been described as "a place where motivation itself is the expertise" [Pre03, p. 1]. Indeed, a game that is not fun to play will not motivate many to play it, thus ultimately causing financial losses for its creators. The question what exactly it is that makes a game fun is difficult to answer and has been discussed in many books on game design (e. g., [Bat04], [SZ03], [Kos05], [Ful08], and [Sch08]) and models of player enjoyment (see Section 3.4.4). In the context of gamification, motivation also plays a crucial role and moving people to exhibit certain behaviors is typically one of its main objectives [HKS14]. To that extent, extrinsic motivators such as points, badges, and leaderboards are often employed to engage users by giving them goals and rewards to strive for (see, e. g., the examples in Section 3.2.1).

A misconception within gamification research is the assumption that intrinsic motivation can be generated from extrinsic motivators if they are properly aligned [ZC11]. However, this is a fallacy, as intrinsic motivation is self-determined and can only lie within the activity itself [RD00b, RD00a]. This means that awarding a person with points for running a certain distance cannot make the runner intrinsically interested in this task. Nevertheless, it has been shown that extrinsic motivators do not always influence intrinsic motivation (neither positively, nor negatively), and that intrinsically motivated users of a gamified systems do not necessarily exhibit a higher task performance [MBTO15]. Therefore, it cannot be said that designers of gamified solutions should only focus on one type of motivation and completely disregard the other, and the focus may depend on the task that is gamified. For instance, [DKR99] reports that extrinsic rewards undermine intrinsic motivation for interesting activities, but not for uninteresting tasks. Consequently, knowledge about how the intended users value the task to be gamified can help with creating an appropriate design concept.

3.5.2 Self-Determination Theory

Self-determination is "a quality of human functioning that involves the experience of choice" [DR85, p. 38]. Not only does the term refer to the *capacity* to choose one's own actions, but to the innate *need* to exhibit such self-determined behavior [DR85]. Consequently, self-determination is connected to acting without external rewards and pressures but for intrinsically motivated reasons. However, Self-determination Theory (SDT) makes a more nuanced distinction between different types of motivation and the degree to which behaviors are self-regulated, ranging from complete amotivation over four types of extrinsic motivation to acting purely for internal reasons [RD00b]. This spectrum is illustrated in Figure 3.17 and consists of the following motivational constructs: *Amotivation* describes the lack of an intention to act, for instance due to not valuing an activity or not feeling competent to do it. *External regulation* denotes behavior performed solely for external reasons, such as rewards or commands. *Introjected regulation* relates to acting out of internal control, for instance to avoid guilt or satisfy one's own pride. *Identified regulation* means that a person acts because they have accepted a behavior as personally important. *Integrated regulation* is the most self-determined form of extrinsic motivation and occurs when a regulated behavior coincides with one's own needs and values. *Intrinsic regulation* describes acting out of internal motivation as previously described. According to Deci and Ryan, the extent to which human behavior is self-

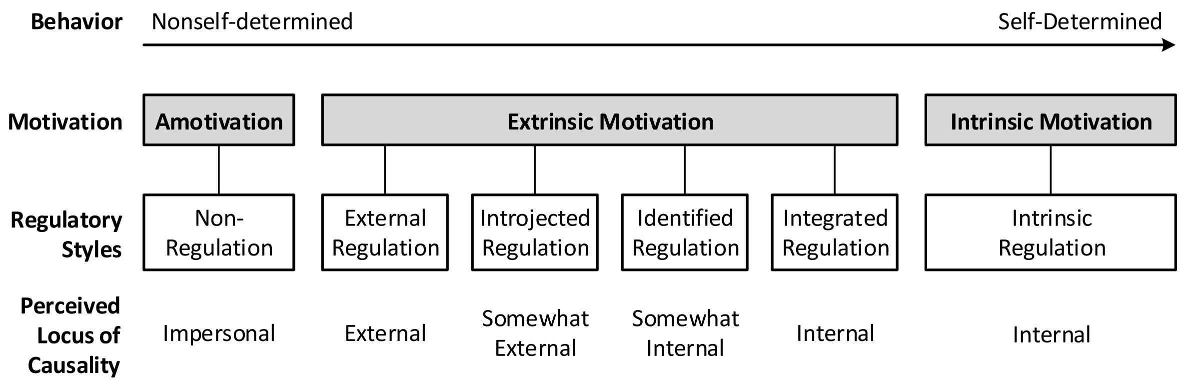


Figure 3.17: Self-determination continuum. Source: based on [RD00b, p. 72].

determined depends on the following three innate, universal psychological needs [RD00b]:

- **Autonomy** is related to "an inner endorsement of one's own actions, the sense that they emanate from oneself and are one's own" [DR87, p. 1025]. Simply choosing to exhibit a behavior is not enough for autonomy, as this might still be done for reasons of control, such as guilt avoidance. Rather, the behavior must be experienced as something for which only oneself is responsible, including the selection of personal own goals and how to achieve them. As mentioned in the previous subsection, extrinsic motivators may undermine perceptions of autonomy [DKR99].

- **Competence** describes "an organism's capacity to interact effectively with its environment" [Whi59, p. 297]. SDT proposes that experiencing enhanced competence, for instance through opportunities to learn new skills, optimal challenge, or positive feedback, support intrinsic motivation by enhancing perceptions of one's own competence [RRP06]. Consequently, SDT is also related to flow and self-efficacy, where feelings of competence play an important roll as well.

- **Relatedness** is the need to "feel belongingness and connectedness with others" [RD00b, p. 73] (also see [BL95]). This is important for the internalization of external motives, as the need to relatedness may move people to perform certain actions because they are valued by others to whom they (want to) feel attached.

In the context of gamification, SDT can make a meaningful contribution by informing designers about the various conditions that support intrinsic motivation and facilitate the internalization of external motives. For instance, Nike+ can support autonomy by allowing its users to set personal running goals. Additionally, the use of such applications in itself is already voluntary, thereby rendering it self-determined behavior. Furthermore, Nike+ allows its users to experience competence through achievable goals that increase over time to illustrate progress and skill development. Lastly, the need for relatedness can also be addressed through social features such as leaderboards and user interaction. More generally, Aparicio et al. present a simple gamification design framework that maps individual game design elements to the three basic human needs [AVSM12]. Specifically, the authors associate profiles, avatars, choice of activities, and various configuration mechanisms with autonomy, positive feedback, optimal challenge, intuitive controls and PBL with competence, and groups, communication features, and social networks with relatedness. However, this framework should be treated with care, as Aparicio et al. do not evaluate their proposed mapping and, as illustrated in [AC11], a simple element such as badges unfold its impacts in many different ways. Overall, it can be summarized that SDT is often invoked in the foundational parts of gamification publications, but very little work carrying these initial thoughts over into implementation and, ultimately, evaluation by means of validated measurement instruments can be found. One positive example is a study carried out by Mekler et al., in which the Intrinsic Motivation Inventory was used to operationalize SDT [MBTO15].

3.5.3 Flow Theory

Flow can be defined as “[a] state in which people are so involved in an activity that nothing else seems to matter; the experience itself is so enjoyable that people will do it even at great cost, for the sheer sake of doing it” [Csí90, p. 4]. A person who is in this state exhibits various characteristics, including “[intense] and focused concentration” on the current activity, “[merging] of action and awareness”, a “[loss] of reflective self-consciousness”, a “sense that one can control one’s actions”, a “[distortion] of temporal experience” and an “[experience] of the activity as intrinsically rewarding” [NC02, p. 90].

Such experiences are rare in everyday life, but can be produced by almost any activity, if the following three conditions are met [CAN05]. Firstly, flow requires a set of clear goals that structure the activity and provide it with purpose and direction. Secondly, flow depends on a balance between the perceived difficulty of the task and the skills a person believes to possess. Lastly, flow is facilitated by the presence of immediate and clear performance feedback. The impact of skill and challenge on flow is also illustrated in Figure 3.18. Here, the "flow channel" highlights the region in which the levels of both match, and thus the flow state can be reached. Beyond this channel, a state of anxiety is entered if the difficulty is too high, and boredom if the skills exceed the challenge. In sum, flow can therefore be related to personal growth and to pushing oneself towards higher levels of performance [Csí90]. However, it is also a highly fragile state that can be easily disrupted when the level of challenge or perceived skill changes. Furthermore, there is no guarantee that individuals enter a flow state even if all conditions are met. Lastly, research indicates a positive correlation between self-determined types of motivation and flow, thereby connecting the latter to intrinsic motivation and SDT [KF99].

The enjoyment that arises out of playing digital games is strongly connected to flow, and today many games are designed with the specific intent to evoke such experiences [Che07]. Consequently, all of the preconditions of flow are also reflected in the game design principles and heuristics that were presented in Section 3.4.3. This can be further substantiated by example of the game Super Mario World that was previously discussed in Section 3.4.2 and has been shown to induce flow experiences [IC07]. In Super Mario World, players are presented with clear goals on multiple levels: immediate goals relating to the currently-visible portion of the screen such as defeating an enemy or jumping over a chasm, level-specific goals such as reaching the exit, and the overall game goal of saving the princess. Furthermore, the difficulty of the game increases over time through the introduction of new mechanics, obstacles, and hazards, and more difficult level design. Lastly, players receive instant feedback, for instance through sound effects when collecting coins or power-ups, or visual effects when players collide with an enemy. Ensuring that these conditions are met throughout the game is a responsibility of the game designer, and by the design decisions that are made, the individual paths of players through (or outside) the flow channel is determined. This is

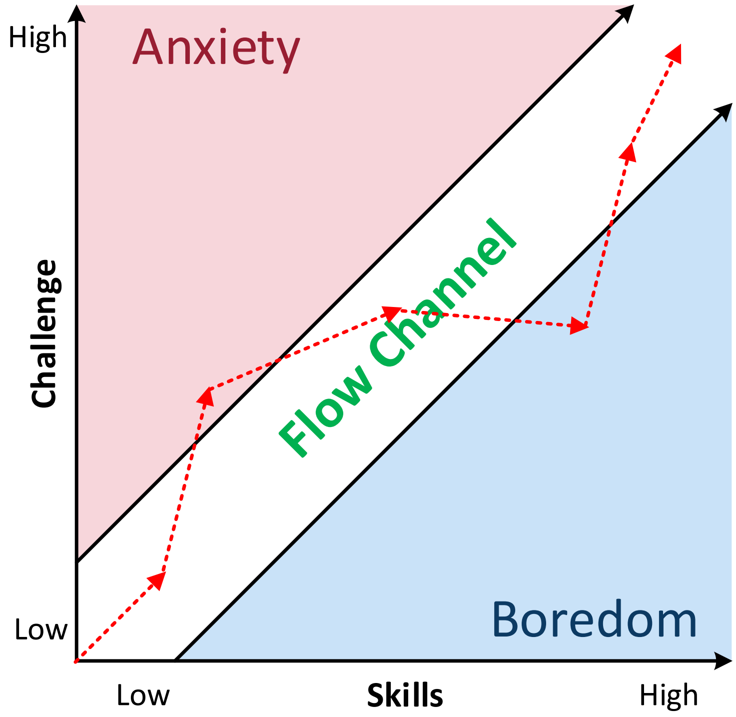


Figure 3.18: Flow channel and possible player journey. Source: based on [Csí90, Che07].

illustrated by the red line in Figure 3.18, which represents the experience of a single player in dependence of gameplay. Research on flow in games has addressed a variety of different topics, e. g., the specification of a model of player enjoyment based on flow [SW05], the design of game design heuristics intended to enable flow experiences [SJW12], and the measurement of flow in games [ZCZ10, OA11, ZFC11].

The capability of flow theory to improve the experience of activities that are not inherently enjoyable was already recognized before the advent of gamification. For instance, in 1990 Csikszentmihalyi suggested that "[mowing] the lawn or waiting in a dentist's office can become enjoyable provided one restructures the activity by providing goals, rules, and the other elements of enjoyment" [Csí90, p. 51]. In that regard, games are a valuable source of inspiration as they hold "valuable principles for making even the most mundane activity more engaging" [Det14, p. 305]. Consequently, the underlying rationale for enabling flow in non-game context lies in examining how games provide clear goals, feedback, and challenges matching the skills of players, and to then transfer these insights to the context of interest, such as physical exercise [HK14] or learning [HSR+16]. According to Blohm and Leimeister, activating flow experiences to increase the voluntary use of gamified products and services and strengthen the motivation and performance of users is one of the main potentials of gamification [BL13]. Therefore, flow has become a concept that is widely discussed by both practitioners and academics [Rac14].

3.5.4 Goal-Setting Theory

A goal is something that "[an] individual is trying to accomplish, the object or aim of an action" [LSSL81, p. 2]. Such aims can for instance be found in performance standards, quotas, norms, objectives, deadlines, or process models [LSSL81]. Goals are motivating because the act of setting them creates a discrepancy between the present condition of a person and an object or outcome, which causes discontent and the desire to attain the latter [LL06]. Furthermore, it has been shown that goals affect an individual's performance at a particular task in four ways [LL02]. First, goals direct effort and attention toward those activities that contribute to the former and away from those that are irrelevant. Consequently, more specific goals have been found to have a higher impact on performance than goals that are rather vague or abstract, such as the directive to simply "do one's best" [LL06]. Second, goals are energizing so that their difficulty is positively correlated with task performance as long as the person is committed to a goal and has the required skills to achieve it [LL06]. Thus, the most difficult goals also result in the highest effort and performance levels. Third, goals impact persistent effort as long as a person has the time the task requires. Last, goals lead to the activation of existing and the discovery of new, task-relevant strategies and knowledge. Similarly to flow theory, goal-setting also requires concrete performance feedback to be effective, because only this allows individuals to adjust the direction or effort or their actions [LL02].

According to McGonigal, goals are one of the defining traits of games that give players a sense of purpose [McG11]. Given a compelling goal and motivating feedback, she argues, players will continuously exert effort until they have exhausted their own abilities and the provided challenge. Consequently, it can be reasoned that goal-setting is also a viable strategy for gamification, and indeed, this theory is increasingly being adopted into the "theoretical canon of gamification research" [Det15, p. 2]. One example is the study conducted by Landers et al. (previously mentioned in Section 3.4.1) that found leaderboards to be implicit sources of goals as users set performance targets aimed towards the top of the ranking for themselves [LBC15]. Another example is a field experiment carried out by Hamari, who introduced badges as goals into a peer-to-peer trading platform and was able to determine a positive impact on various measures of user activity [Ham17].

3.5.5 Self-Efficacy Theory

The concept of perceived self-efficacy relates to a person's belief that they can successfully carry out an activity required to produce a certain outcome [Ban77]. The stronger this belief, the more effort people will exert and the longer they will persist in a behavior despite difficult obstacles and experiences of failure [Ban82]. Conversely, doubts about their own capabilities may lead individuals to expend less effort or cease an activity altogether. Therefore, self-efficacy has a strong influence on the behaviors that people exhibit, the challenges they undertake, and their reaction to intimidating situations. It must also be stressed that beliefs of personal efficacy are not a global trait, but rather depend on individual characteristics and the nature of an activity [Ban06]. Bandura notes that expectations of self-efficacy are mainly derived from the following four sources of information [Ban77]:

- **Performance accomplishments**: Previous experiences of mastery increase perceived self-efficacy, whereas repeated failures have a negative impact. The tolerance for failures may increase once a strong belief in personal efficacy has been developed.
- **Vicarious (indirect) experience**: Perceived self-efficacy may be positively affected by observing others perform difficult activities without negative consequences. This leads to the expectation that oneself can improve too through persistent effort.
- **Verbal persuasion**: People can be led to believe that they are capable of overcoming challenges that have previously overwhelmed them through

verbal suggestion. While the impacts of this on efficacy beliefs are likely to be small, verbal persuasion is easily administered and readily available in most situations.

- **Emotional arousal**: Stressful, challenging situations may also cause emotional arousal that can have an impact on perceived self-efficacy.

Self-efficacy theory is strongly related to many of the other motivational concepts that are discussed in gamification literature. For instance, experiences of mastery as considered in SDT may lead to higher efficacy expectations. Furthermore, a match between the perceived difficulty of a challenge and the skill required to overcome it (i. e., the perceived self-efficacy), is also an important precondition for the flow state. Finally, the positive impacts of difficult goals on task performance predicted by goal-setting theory also depend on self-efficacy—and inversely, self-efficacy influences the goals that individuals set for themselves [LL06]. Consequently, designers of a gamified solution must keep in mind that individuals may exert varying levels of effort to accomplish the gamified activity in dependence of their perceived self-efficacy. For instance, if they believe themselves to be incapable of unlocking a particular badge, its goal-setting function will be undermined.

3.5.6 Technology Acceptance

Research on technology acceptance is concerned with a central problem of the IS discipline: gaining an understanding of why humans accept or reject IS (typically in an enterprise context) to “better predict, explain, and increase user acceptance” [DBW89, p. 982]. One of the most influential theories in this context is the Technology Acceptance Model (TAM), which was proposed by Davis as an extension of the Theory of Reasoned Action (TRA) [AF80] and has the primary goal of explaining the determinants of continued IS use [DBW89]. To that extent, it proposes that the beliefs of an individual about the *perceived usefulness* and *perceived ease of use* of a computer system are of central importance for its acceptance and subsequent use [DBW89]. These two constructs are defined as “the degree to which a person believes that using a particular system would enhance his or her job performance” [Dav89, p. 320] and “the degree to which a person believes that using a particular system

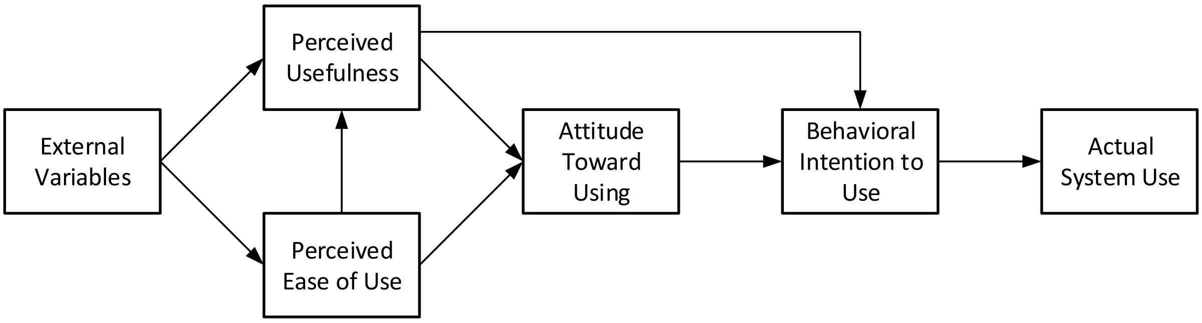


Figure 3.19: Technology Acceptance Model. Source: based on [DBW89, p. 985].

would be free of effort" [Dav89, p. 320], respectively. It should be noted that the latter is again related to self-efficacy theory. The constructs and relationships defined by the Technology Acceptance Model (TAM) are illustrated in Figure 3.19. As shown, certain external factors, such as system functionality or usability features, influence the beliefs of users about usefulness and ease of use. Furthermore, perceived ease of use also has an impact on usefulness beliefs about a system. Both influence an individual's attitude toward the continued use of an IS, which together with perceived usefulness yields a concrete behavioral intention. Ultimately, this results in actual, observable system use. Since its inception, the Technology Acceptance Model has been extended with a variety of additional factors to increase its explanatory power, such as subjective norms, anxiety, or playfulness (see, e. g., TAM 3 [VB08] and the Unified Theory of Acceptance and Use of Technology (UTAUT) [VMDD03]). Furthermore, despite its widespread adoption, the TAM is not uncontested and has become the target of considerable critique [Bag07].

In the context of gamification research, the Technology Acceptance Model and its extensions are of considerable relevance as they may support academics with explaining *how* and *why* gamification works exactly [PT15]. For instance, Hwang found that the perceived enjoyment (i. e., intrinsic motivation [Ven00]) of employees using an Enterprise Resource Planning (ERP) system had a significant impact on the perceived usefulness and ease of use of the latter [Hwa05]. The positive impact of intrinsic motivation on perceived ease of use was also determined in another study [Ven00]. More generally, a recent meta-analysis of 303 studies concluded that for *utilitarian IS* (productivity-oriented, mostly used in the workplace), extrinsic motivators are more important for continued use, whereas for *hedonic IS* (pleasure-oriented), the role of intrinsic motivators is more crucial [WL13]. These insights can help gamification designers with the selection of game design elements depending on the expected usage motives of the solution to create. In particular, gamification postulates that the result is used for a primary purpose other than enjoyment [DDKN11], which indicates that extrinsic motivators should not be directly discounted as an “inferior” source of motivation.

3.6 Criticism

Similarly to other recent technological developments such as Big Data or Cloud Computing, gamification quickly became a widely-used buzzword generating a considerable amount of hype. For instance, market research firm Gartner has formulated strongly optimistic prognoses such as “by 2015, more than 50 percent of organizations that manage innovation processes will gamify those processes” [11] and “by 2015, 25 percent of all redesigned processes will include one or more gamified engagement practices” [14]. This was accompanied by an introduction of gamification into the Gartner hype cycle for emerging technologies close to the “peak of inflated expectations” in 2011 [12] where it stayed in 2012 [15] before climbing to the peak in 2013 [16]. Furthermore, practitioners added to the hype by exaggerating the potentials of gamification, for instance through the promise that it “presents the best tools humanity has ever invented to create and sustain engagement in people” [ZL13, p. xvii] to “vastly improve the world as we know it - and deliver the organizational success you desire” [ZL13, p. xvii]. However, a certain degree of disillusionment followed when gamification failed to fully deliver on its promise. Consequently, Gartner revised its original prediction, now stating that “by 2014, 80 percent of current gamified applications will fail to meet business objectives primarily because of poor design” [13]. Similarly, many practitioners have become more careful by highlighting that the “true” potential of gamification must be sought beyond the hype that surrounds it (e. g., [WH12], [Bur14]). To illustrate the underlying reasons for this disillusionment, the following paragraphs will summarize some of the most important points raised by critics of gamification.

As mentioned at the beginning of this chapter, gamification is an industry-driven trend, meaning that a considerable part of its perception in the public eye was shaped by early deliberations on the phenomenon by practitioners and consultants [8]. Soon after the term became popular, many game designers and game studies researchers positioned themselves as gamification critics (e. g., [32], [4], [30], [31]), some even going as far as dismissing gamification entirely as “bullshit” [3, Bog14]. This already demonstrates that the discussion about gamification is highly emotional and often involves extreme positions and considerable cynicism. Generally speaking, critics take offense at how, in their view, practitioners misunderstand what makes games fun to play, yet try to use their power to exploit people (e. g., [7], [31]). Much of the criticism of gamification is directed towards the works of Gabe Zichermann (e. g., the book review by Deterding [7]), the “dark lord” of gamification [4], whose book *Gamification by Design* [ZC11] was released in the early stages of the trend and is thus seen as standard reference for gamification by many (cf., [8]). Accordingly, Google Scholar lists it as the second-most cited publication on the topic directly after the seminal article by Deterding et al. [DDKN11].

As Ian Bogost appropriately puts it, “[making] games is hard. Making good games is even harder” [4]. Unsurprisingly, only a fraction of the games that are released every year are actually considered good games by critics and players. For instance, an examination of the review aggregator Web site *metacritic* reveals that the average score of all Xbox One games released to date is 70.19%[3]. Generally speaking, making good games is a costly endeavor, with the recent blockbuster *GTA V* as an extreme example requiring five years, a budget of £170 million, and a core team of 360 people to be developed [10, 22]. Thus, following the logic of Bogost’s continued statement that “[making] good games that hope to serve some external purpose is even harder” [4], it becomes clear that creating a good gamified application is difficult as well and requires time, money, and expertise. However, these investments are made only seldom in practice, one of the main reasons for this being that gamification as it is marketed by practitioners must be “easy” [3]. Following this logic, it needs to offer simple, repeatable approaches so that it can be packaged and sold by consultants and startups with ease and limited marginal cost [4]. For that reason, gamification typically relies on simple design patterns in lieu of proper game design methods, as the former provide easily applicable solutions whose

[3] See: http://www.metacritic.com/browse/games/release-date/available/xboxone/metascore. Last accessed: 2016-11-25.

effectiveness is commonly accepted in the context of games. This is done following an *additive* approach that assumes an uninteresting activity can be made enjoyable if enjoyable elements (taken from games) are added to it [Det15]. Thus, Herger states that gamification practitioners "most often bolt gamification onto a system" [Her14, p. 27]. However, an examination of real-world gamification applications quickly reveals that this is not globally true. For instance, all three of the examples presented in Section 3.2.1 are based on activities that might be of inherent interest to many individuals (e. g., running or learning a language), and include game design elements as central components rather than shallowly-integrated additions.

Indeed, the game design elements most commonly discussed by practitioners as well as academics are the interface design patterns *points*, *badges*, and *leaderboards* [WH12]. These three elements are used in conjunction to such an extent that they have received their own abbreviation (PBL) and are also referred to as the *blueprint* of gamification [Det14]. However, as discussed in Section 3.4, points, badges, and leaderboards sit at the lowest level of game design elements and are merely used to provide feedback and communicate progress to players. As such, they do not relate to actual gameplay, do not create the same experiences that games do, and do not represent unique characteristics of games [4]. Consequently, as Robertson points out, PBL-based gamification means "taking the thing that is least essential to games and representing it as the core of the experience" [32]. Indeed, all three elements can be found in many other domains unrelated to games. For instance, points are used in loyalty programs, credit ratings, education, and product testing, badges as records of achievement can be found in boy scouting and the military, and leaderboards are employed in sporting competitions. Thus, it stands to reason that any of these other areas might have served as an alternative inspiration for gamification and that PBL does not, in fact, relate to games at all. Consequently, Robertson suggests that a more appropriate name for this kind of gamification would be *pointsification* [32].

Ultimately, all of the previous points yield the final result that gamified systems are seldom more enjoyable or fun to use than their non-gamified counterparts, which directly contradicts one of the most important goals of gamification. The reason for this can be found in a closer examination of how points, badges, and leaderboards are actually used in the context of gamification: as *rewards* that the users can earn for performing certain actions that are deemed desirable (e. g., [ZC11]). However, as Radoff argues, reward delivery systems and the associated feedback loops, while undeniably important for games overall, are only a small part of what makes games work the way they do [30]. Deterding expands upon this point by describing a fictional game in which the player earns one trillion points every time he presses a button (see Figure 3.20) [7]. Assuming that rewards are the main source of fun and enjoyment in games, this example should be one of the most enjoyable games possible, which is clearly not the case. Instead, the pleasure resulting from playing a good game ultimately comes from mastering challenges that are interesting rather than tedious, such as solving a puzzle, jumping across a large chasm, or defeating a difficult monster [32, 4, 7]. However, this is not to say that gamification by means of extrinsic motivators cannot yield positive results, as the examples presented in the previous subsection illustrate. Nevertheless, some researchers have appealed for gamification to be re-conceptualized as a process of crafting *gameful experiences* rather than simply using elements from games (see, e. g., the books *Rethinking Gamification* [FFRS14] and *The Gameful World* [WD14]). This understanding is most strongly reflected in the definition of the term proposed by Werbach in [Wer14]. Consequently, a gamification endeavor shall be considered successful if the use of the gamified system leads to similar experiences as playing a game.

Gamification practitioners and researchers often refer to models and theories from game design and psychology to explain why gamified systems work and how they should be designed. Popular examples include extrinsic/intrinsic motivation, self-determination theory, and flow theory (cf. Section 3.5). However, a closer examination of gamification literature reveals that these foundations are often misunderstood or misused, thereby calling any insights built on them into question. For instance, Burke states that gamification can be distinguished from reward programs by a focus on intrinsic rather than extrinsic motivators, but later builds his design recommendations around points, badges, and leaderboards, which are prime examples of extrinsic incentives [Bur14]. Similarly, Zichermann reduces the concept of flow to a state that is reached when a match between skill level and the difficulty of a challenge exists, which is only one of the many preconditions that must be met for a possibility of flow to occur [CAN05]. Furthermore, Deterding notes the belief

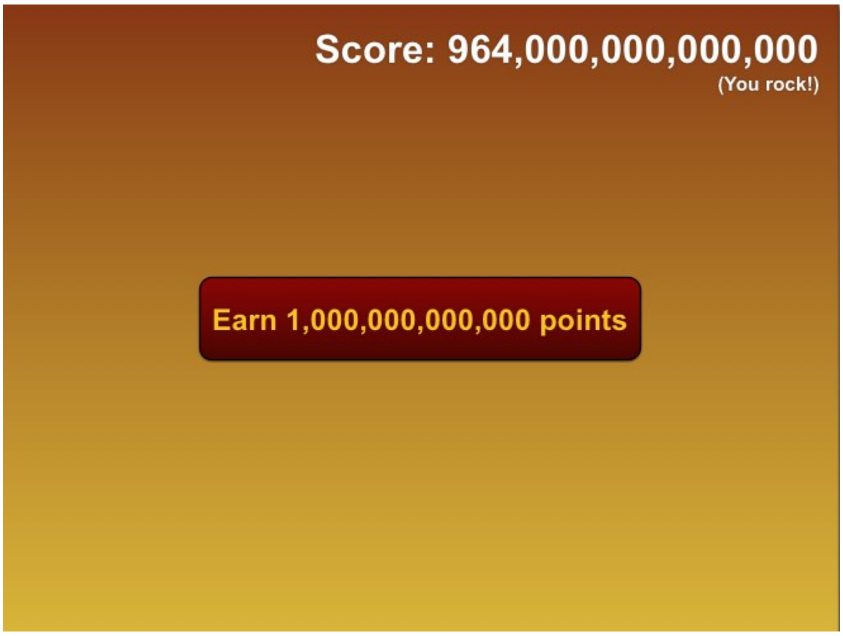


Figure 3.20: Sample screenshot of a button-pressing game. Source: [7].

held by many authors that there is a deterministic one-to-one relationship between the use of a certain game design element and the motivational effect it creates [Det15]. However, depending on personal characteristics, context, and other factors, a simple element such as feedback could either enhance intrinsic motivation if it is perceived as supporting competence, or decrease intrinsic motivation if it is perceived as diminishing autonomy [Det14]. As these misunderstandings form the foundation from which design decisions for gamification projects are derived, their potential harmfulness should not be underestimated. Another issue lies in the fact that gamification consultants frequently propose their own, unsubstantiated theories, presumably in an attempt to sell their own consulting services [Det15]. For instance, Marczewski combines SDT [DR85] with the work by Daniel Pink [Pin11] to create his own model of intrinsic motivation—Relatedness, Autonomy, Mastery, Purpose (RAMP)—with an abbreviation that should most likely work as a convenient visual metaphor for success [20, Mar15]. Other examples are the SAPS (Status, Access, Power, Stuff) model of rewards proposed by Zichermann [ZC11] and the Octalysis gamification framework by Chou [Cho15].

A final, substantial cluster of criticism is directed towards the purpose and ethics of gamification. In the words of danah boyd [sic], gamification is "a modern-day form of manipulation" [1] that can both help people and hurt people. For Zichermann, the focus seems to be on the latter, as he describes replacing real rewards that have a monetary value with virtual rewards whose real value cannot accurately be determined by customers as one of the main benefits of gamification for organizations [ZC11]. Thus, it is hardly surprising that the most vocal critics of gamification claim that its *raison d'être* is to trick its users into doing work for free (e. g., [4, 6, 31]). Accordingly, "exploitationware" [4] and "playbor" (an amalgamation of the words "play" and "labor" [31]) have been suggested as alternative denominations for the phenomenon. This practice is deemed unethical, as participants create actual value, yet do not come to possess a significant portion of this value [31]. Instead, the activity of creating value becomes the reward in itself by being enjoyable and fun. This, Chaplin argues in turn, is problematic, because it may divert attention from actual problems that exist in real life by obscuring them through simulated feelings of satisfaction [6]. Nevertheless, it should be noted that gamification is not only used in business contexts, but also for non-profit purposes such as those presented in Section 3.2.2.

Part II

Gamification for Business Process Modeling

4 Problem Specification

Early research on Business Process Management has had a strong focus on technological issues such as modeling tools and techniques, standards, workflow execution engines, workflow patterns, and service-oriented architectures. Even today, a considerable portion of research published in relevant outlets for this field, e. g., the BPM conference or the Information Systems journal, focuses on issues that have little connection to the humans who are involved in business processes or affected by their outcomes. A prominent example for this is *process mining*, a topic that is concerned with the extraction of process descriptions from the traces of enacted processes stored in log files [vdARS05]. This is an inherently technical problem that considers humans only peripherally. However, as the six core elements of BPM proposed by vom Brocke et al. highlight, managing processes consists of more than merely methods and IT, and properly considering the people involved in managing business processes is an essential requirement for these efforts to be successful.

In recent years, academics and practitioners have started to include BPM stakeholders in their efforts as evidenced by a variety of different developments. For instance, many tool vendors, such as Horus[1] and Signavio[2], are nowadays incorporating functionality to support collaborative, distributed process modeling projects into their modeling tools. In addition, the market research firm Gartner has coined the term intelligent Business Process Management System (iBPMS) to describe and rank such BPM systems that provide an enhanced support for collaboration, social media, and other types of interaction [SSHJ12]. In the context of academic research, the rising importance of people-centric issues can be observed along (at least) three dimensions. Firstly, researchers have started to address new research topics such as the importance of cultural values for BPM [vBS11, SvBR13], the role of gender [GSSM16], or how novice

[1] See: http://www.horus.biz/. Last accessed: 2017-03-20.

[2] See: https://www.signavio.com/. Last accessed: 2017-03-20.

modelers create business process models [RSR10, RSR12], that are inherently related to humans. Secondly, there is an increasing number of outlets that allow for the placement of human-centric BPM research, such as the Workshop on Business Process Management and Social Software[3] (BPMS2) at the annual BPM conference or the Business Process Modeling, Development, and Support (BPMDS) conference[4]. Lastly, some authors are also proposing new "flavors" of BPM placing humans into the center of the discussion, such as collaborative BPM [NH11, NP11], Social BPM [EGH^{+}10, PV14], and BPM 3.0 [SK09].

The main purpose of this chapter, which is based on previous work by the author [PV13b, PV14], lies in highlighting the problems that BPM is currently facing that can be addressed through gamification. To that extent, Section 4.1 first presents Social BPM as one of the most inclusive approaches to BPM that is based on an empowerment of all relevant process stakeholders. Section 4.2 then provides a discussion of the challenges that Social BPM entails, which is followed by an illustration of the most significant contributions that gamification is hypothesized to be able to make in Section 4.3. Next, an overview of related work is provided in Section 4.4. Based on these deliberations, an outlook on the remainder of this thesis is presented in Section 4.5.

4.1 Social Business Process Management

Traditionally, Business Process Management is understood as the domain of experts; each of the individual activities envisioned in the BPM life cycle is carried out by a well-defined set of specialists (e. g., process modeling by modeling experts, implementation by IT experts, and enactment by process end-users), and interactions between these groups are the exception rather than the rule. This is especially true for process modeling, which is seen as a task carried out by a small number of highly-trained experts eliciting process requirements from other stakeholders through interviews, workshops, and meetings [FMP11, RMH13]. This can be seen as a top-down approach in which those individuals who are eventually responsible for executing business processes are not empowered to make direct contributions to their improvement based on knowledge and experiences gained from past enactments [SR12].

[3] See: http://www.bpms2.org. Last accessed: 2017-03-20

[4] See: http://www.bpmds.org. Last accessed: 2017-03-20

However, as the knowledge about current business practices and how they can be improved is distributed among a potentially large, heterogeneous set of process stakeholders, this does not properly account for corporate reality [RMH13]. Therefore, it is hardly surprising that BPM literature offers a vast wealth of publications emphasizing the importance of actively involving a large variety of different stakeholders to ensure their acceptance of the implemented solutions [AF08, BGR05, BICS07a, BKR03, BNP10, Gil10, RB10, Ros06a, Ros06b, RvB10, SK09, ST13, Trk10, vBBB^{+}11, vBSR^{+}14]. Additionally, since business processes are increasingly blurring the boundaries between separate organizational entities, BPM must also account for the perspectives of external stakeholders such as customers and suppliers to be successful [NP11]. If these issues are disregarded, the following problems may occur [EGH^{+}10, SN09]:

- **Model-reality divide**. The model-reality divide can occur when process end-users reject implemented business processes because they do not accurately reflect their day-to-day work activities. Thus, the term describes a state of divergence between processes as they have been planned and as they occur in reality, i. e., between idealized process models and process enactment. Possible reasons for this phenomenon are the insufficient consideration of process stakeholder knowledge and a lack of flexibility and responsiveness of BPM.
- **Loss of innovation**. In organizations employing "traditional" BPM, employees may choose not to share their ideas for process improvement and innovation, as they perceive the guidelines for process change management as too restrictive and intransparent and consider the chance of success as too small. Thus, this knowledge is either only applied locally at the level of individual process instances, or lost entirely, which may further increase the model-reality divide.

To overcome these problems, the management of business process models must be re-conceptualized as a task for which the *entire* business community assumes responsibility rather than just a few individuals [SR12]. This may be achieved by means of *social* BPM, which invites a large number and variety of internal and external stakeholders with varying skill sets and degrees of BPM proficiency to contribute their domain and method expertise. Consequently, all participants are "promoted" from being passive consumers of information to active producers of BPM artifacts. In practice, Social BPM can manifest itself in two forms: *social business process engineering* and *social business process execution* [Kem11, Ric11, SVOK12]. Whereas the former denotes the practice of involving all process stakeholders in the design and improvement of business processes through a suitable technical and organizational environment, the latter is concerned with making process enactment more social by allowing the execution of process activities by individuals not specified *a priori* at design-time [BFV12], for instance by invoking crowdsourcing platforms [KTD+13]. In both cases, by working together, the business community is enabled to leverage its "crowd wisdom" [Sur05] to work towards creating better, higher-quality solutions for supporting business processes than any single expert would be able to achieve alone. To create this "architecture of participation" [EGH+10], Social BPM must be implemented through "the involvement of all relevant stakeholders in a BPM life cycle by applying social software and its underlying principles" [PV14, p. 3870]. Following this definition, Social BPM is characterized by the following two properties:

Utilization of social software. From a purely technological perspective, Social BPM is characterized through the use of various types of social software within the activities of the BPM life cycle, thereby adding social functionality to traditional Business Process Management System (BPMS) [BJP11, DC11, Puc11, SN10]. Social software is a class of Web-based applications that support humans with exchanging information, establishing relationships, and communicating in a social context [Hip06]. Its main purpose lies in enabling workflows for the generation of digital artifacts that combine the contributions of numerous individuals who do not necessarily know each other and who are—at least initially—organized in a non-hierarchical fashion [SN09]. Consequently, social software has also been called "software that gets better the more people use it" [VH07]. Concrete uses for social software discussed in the context of Social BPM literature include the following:

- **Wikis** are a collaborative content authoring tool whose (possibly anonymous) users have equal viewing and editing rights and are not organized in any predetermined fashion. This allows for an exhaustive documentation of topics of interest by increasing the availability of knowledge only held by a small number of experts on top of common knowledge

[KR09]. In the context of Social BPM, Wikis have been proposed as tools for modeling and documenting business processes [DV11, GRS12, SR12], as well as executing and managing individual process instances [NE09].

- **Tagging**. Tags are descriptive terms that users can freely assign to specific (abstract or real) objects to provide details about their semantics. There are no restrictions on which terms can be used and what their purpose should be. Furthermore, tags are shared, can (and should) be assigned by all users, and taken together they form a flat, non-hierarchical classification also referred to as a *folksonomy* [Pri10, VFL10]. Their use has been proposed to improve the search for tagged models or model fragments within larger model repositories [Pri10, SMM$^+$10], or for using tag-based similarity metrics to find similar models [LB13].

- **Social recommendations** exploit the data provided by a community to suggest content to users based on, e. g., how similar their tastes are to others. In the context of Social BPM, this can be used to provide users who are unsure about how to create, extend, or modify a process model with fitting suggestions. To that extent, modeling suggestions can be derived from the metadata and contents of already-existing models, the past modeling behavior of focal users and other individuals, and the relationships between them [KSR09, SWMW09].

- **(Micro-)Blogs**. In their traditional manifestation, blogs allow individuals to create a chronologically-ordered journal of professional or private matters. Typically, readers can subscribe to updates of blogs (so-called feeds), which allows aggregating multiple journals in a single space. This facilitates remaining up-to-date about large numbers of blogs without having to access each of them individually. Microblogs are similar in nature, but impose significant restrictions on the number of textual characters that can be posted [KR09]. In Social BPM literature, blogs and microblogs are proposed as tools that can be used to broadcast and monitor the status of process instances in execution [Böh11, NE09, VF12].

- **Social networking sites** allow users to establish and maintain relationships with other individuals and communities of interest. Features of such applications include creating and updating personal profiles, defining interpersonal connections, communicating with others, creating topic-oriented groups, and sharing content [KJL10]. In the context of Social BPM, social networking sites can facilitate searching for partners for the collaborative enactment of process instances [Bra13, DV11], and giving and receiving feedback on process models [HPS13].

Application of the principles of social software. It is a widely-accepted fact that investments in IT often require complementary organizational changes to be successful [BH98, DGK03]. Therefore, simply installing social software does not necessarily guarantee that the full benefits offered by Social BPM can be reaped. To increase the likelihood of success, practitioners suggest that the underlying principles of social software must be adopted as well, which may in turn drive the aforementioned organizational changes. In relevant literature, social software is typically characterized by a combination of the following properties (cf. [SN09, SN10, VFL10]):

- **Self-organization**. A system is called self-organizing if it can ensure and refine its functioning without external influences through the cooperation of its individual components [VFL10]. This means that Social BPM is conducted and governed in a bottom-up fashion that emerges naturally from the way the business community cooperates rather than being mandated top-down (cf. [Hip06]). Should any conflicts arise, Social BPM aims to resolve them at the hierarchy level at which they occur rather than via escalation [MJ10]. As corporate reality may often not be able to accommodate truly bottom-up Social BPM, it may be necessary to find a compromise between the former and more traditional BPM [SR12].
- **Egalitarianism**. Within the limits of reason, all users of a social software possess equal rights and can thus not only view, but also edit all digital contents, including those authored by other individuals [JAW09, SN10]. This holds independently of the organizational membership, skills, or other personal characteristics of an individual [BDJ+11]. For Social BPM, this means that activities such as process modeling are not only performed by experts, but also by the process end-users themselves. More generally, any actor who holds certain knowledge from which

the Social BPM community can profit should have the possibility and capability to make this knowledge explicit in any phase of the BPM life cycle. This entails a convergence of the roles of domain expert and method expert.

- **Collective intelligence**. Many types of social software are based on the assumption that the collective wisdom of a community allows for the creation of superior solutions for a specific problem than individual experts could create by themselves [SN09, Sur05]. In this context, so-called *weak ties*—sporadic connections that only exist for brief periods of time—are of special concern, as they make the “long tail” of knowledge and innovations that would not be captured otherwise available [MWBR12]. For Social BPM, this implies that an organizational environment and culture fostering continuous participation and contributions by all process stakeholders must be established to leverage their collective intelligence.
- **Social production**. Through social software, individuals create two types of artifacts, both of which are considered valuable [EGH^+10, SN09]: *content* such as texts, multimedia data, and diagrams, and *context* information such as social links, reputation, and annotations. Through expansion, modification, and selection, new artifacts are continuously evaluated so that only the best available ideas “survive”. In the context of Social BPM, these ideas are related to the business practices of an organization and how they can be improved. As new information becomes visible and effective immediately, an agile cycle of process improvement without disruptive delays in the transfer of a desirable change to practice is enabled [EGH^+10, Rus11, SN10].

By applying the principles of social software to BPM, organizations can establish an “architecture of BPM participation” that encourages and enables all relevant process stakeholders to actively and continuously participate in process management. Consequently, they are allowed to contribute their individual knowledge and expertise to work towards closing the model-reality divide and improving the organization’s capability to innovate. This does not only provide advantages to the enterprise using Social BPM, but can also improve the work life of each individual employee. However, Social BPM is not without its own challenges, which are discussed in the following section.

4.2 Challenges of Social BPM

Due to its conceptualization around the characteristics of social software, successful Social BPM engages a much larger and heterogeneous set of process stakeholders to increase the quality, quantity, timeliness, and variety of contributions to BPM solutions. This imposes various technical and organizational challenges on BPM, particularly in relation to the inclusion of novice users (e. g., domain experts, process end-users) in BPM activities. Focusing on the six core elements of BPM described in Section 2.4, this means that most challenges arise out of the consequences Social BPM has on the factor “people” either directly or indirectly via other core elements. A literature-based overview of the specific challenges of democratized, bottom-up process management is provided by Pflanzl and Vossen in [PV13b] (for the core element “people”) and [PV14] (for all six core elements) and summarized in Table 4.1. In the following sections, three of these challenges will be discussed in detail, namely “ensuring model quality” in Section 4.2.1, “motivating participation” in Section 4.2.2, and “educating and training participants” in Section 4.2.3. As it will be argued in Section 4.3, it is these issues where the application of gamification can make the most significant contribution.

4.2.1 Ensuring Model Quality

The quality of a business process model is a multi-dimensional property that directly impacts its ability to serve its intended purpose. This is especially true for process models used in the context of IS development, and, as Moody puts it, “[while] a good conceptual model can be poorly implemented and a poor conceptual model can be improved in later stages, all things being equal, a higher quality conceptual model will lead to a higher quality information system” [Moo05, p. 245]. Looking at real-world process models however, it can be seen that maintaining a high quality level is not a trivial task. As illustrated by the sample studies in Table 4.2, process models often contain semantic errors such as deadlocks, are not well-structured, have labels deviating from suggested styles, make improper use of OR connectors, are excessively large, or have layout issues. Many of these problems escalate with increasing model size: the larger a model, the higher its chance to contain a formal error [MVvD+08],

Table 4.1: Challenges of Social Business Process Management. Source: [PV13b, PV14].

Challenge	**Description**
Involving external stakeholders	Select and involve process stakeholders from external organizations in Social BPM.
Motivating participation	Motivate process stakeholders to continuously invest their time and effort to participate in Social BPM.
Educating and training participants	Provide measures through which novice users can build up BPM skills to improve their participation.
Providing modeling tools for novices	Facilitate the participation of novice users through appropriate BPM software and modeling languages.
Ensuring model quality	Maintain a high quality of process models and other artifacts despite the participation of novice users.
Handling information overload	Ensure that participants can find contents that are relevant for them and to which they can contribute.
Integrating semantics	Handling the language gap that arises from the involvement of participants with different backgrounds.
Modeling social enactment	Provide modeling languages that can depict scenarios with social business process execution.

with the error probability reaching 50% at over 50 model elements [MRvdA10]. Inversely, smaller, more modular process models (i. e., models employing hierarchical abstraction and subdivision into sub-processes) have a lower error probability and higher understandability [RM08]. This is problematic, because many real-world process models are in fact very large despite being highly modular. For instance, [RS15] mentions a sample model with 1.154 nodes and 102 sub-processes reaching as deep as 7 levels of abstraction. Due to the fact that the *raison d'être* of process modeling is to cope with the high complexity of real-world business processes, overly complex business process models can be seen as an inappropriate contradiction [BRvU00].

In traditional process modeling projects, stakeholders can be subdivided into two distinct groups: *domain experts* with superior knowledge about the object that is under consideration, and *method experts* that are highly proficient in the use of modeling languages and tools and possess sophisticated abstraction skills [RFME11]. In Social BPM, this subdivision can become arbitrarily blurred, so that individuals without modeling skills may become process modelers. This is problematic, as maintaining a high model quality in the presence of actively-involved novice modelers has been recognized as a difficult task [Ros06a, EGH+10, MRvdA10, FL15]. At the same time, understandability becomes increasingly important as novice users are forced to interpret process models with much less experience and a smaller skill set [BRvU00]. Therefore, the presence of tools that allow inexperienced modelers to operationalize simple rules—for instance the 7PMG—for the creation of good models is of special importance for Social BPM. However, despite the existence of business process modeling tools, modelers still receive very little support with the creation of understandable diagrams [MRvdA10]. Other measures that can help novice users to work with process models include simulation, animation, and explanation generation [KJ03].

4.2.2 Motivating Participation

One important ingredient for Social BPM are the stakeholders who should contribute their knowledge and expertise to process management. Thus, such an approach can only be successful if all individuals who can make relevant and meaningful contributions are sufficiently motivated to invest their effort

Table 4.2: Empirical studies examining quality defects of process models.

Source	Details
[GL07]	Subject of analysis were 285 process models from various sources, including theses, papers, textbooks, and reference models. Models were examined regarding the proper use of OR connectors according to a style rule defined by the authors. Results indicate that 38% of all models make improper use of OR connectors.
[Men09]	Subject of analysis were 2.741 process models from 7 model sets with reference models and models from practice, academia, and books. Models were examined regarding soundness, in particular deadlock-freeness and proper synchronization. Results indicate that 14% of all models contained an error and a large variation between model sets ranging from 0% to 38% faulty models.
[MRR10b]	Subject of analysis were 604 process models with 19.838 activity labels from 1 model set. Models were examined regarding the adherence of activity labels to the proposed verb-object style. Results indicate that only 60% of all labels followed the proposed style.
[LESM+13]	Subject of analysis were 1.241 English, 445 German, and 59 Portuguese models from 3 model sets. Models were examined regarding the adherence of activity labels, event labels, and gateway labels to specific label styles. Results indicate high degrees of deviations from the proposed styles for all types of labels.
[PLM15]	Subject of analysis were 2.498 process models from 3 model sets with reference models, and models from practice and academia. Models were examined regarding lexical ambiguity of labels caused by the use of synonyms and homonyms. Results indicate many occurrences of such ambiguities, e. g., 191 uses of the homonym *to process* and 173 uses of the synonyms *to check* and *to control*.
[RS15]	Subject of analysis were 174 industrial process models with 1.262 sub-processes. Models were examined regarding soundness, in particular deadlock-freeness and proper synchronization. Results indicate a total number of 2.428 errors in the examined models.
[LMG16]	Subject of analysis were 585 process models from 6 companies. Models were examined regarding a total number of 35 quality guidelines and rules for correctness relating to structure, layout, and labeling taken from textbooks. Results indicate high percentages for some errors, e. g., sub-processes inconsistently connected to their main process (>80%), deadlocks (>20%), excessive model size (>45%), overlapping nodes and edges (>25%), and labels deviating from recommended styles (40%).

and time lastingly and continuously to create, modify, and maintain process models and other artifacts. Whereas having only a small amount of highly-active contributors may be an acceptable scenario in some cases, this would contradict the principles on which Social BPM is based, and thus pose a risk to avoiding lost innovation and overcoming the model-reality divide [EGH+10].

To achieve continuous participation, a *critical mass* of users and contents must first be reached, as simply providing process stakeholders with the technical infrastructure and organizational environment that Social BPM requires will not necessarily lead to its adoption if there is a lack of meaningful initial content. Thus, it is advisable for organizations to perform a careful selection of key users who are given incentives to create the first BPM artifacts and promote Social BPM to others. This stage of the introduction of social software is also called “ramp-up” phase, and until it has been completed, the effectiveness of Social BPM will not be able to reach its full potential [EGH+10].

Once Social BPM has been successfully introduced, it must be ensured that the motivation of participants to stay involved with process management activities and continuously contribute their newest skills, knowledge, ideas for innovation to the BPM community is sustained. This can be difficult, as employees may perceive the involvement in Social BPM as time consuming, yet participation should occur on a voluntary basis rather than being forced [EGH+10]. To solve this problem, organizations can either aim to make individuals interested in the BPM activities themselves (i. e., facilitate intrinsic motivation), or to provide them with extrinsic motivators such as tangible rewards, monetary gratification, or other positive (or negative) consequences based on some performance indicators.

4.2.3 Educating and Training Participants

Two types of factors influence whether a reader is able to comprehend a process model: *model characteristics* such as the complexity metrics discussed in Section 2.7.3, and *personal factors* including modeling experience and education [MRC07, RM11]. Due to the inclusive nature of Social BPM, the set of prospective contributors not only includes individuals with a well-founded knowledge of BPM methods and tools, but also less experienced users with a limited skill set. This poses a challenge for Social BPM, as the model-reality divide can only be closed and ideas for process innovation are only captured if every process stakeholder is enabled to explicate their domain knowledge via process modeling. However, experts generally deny modeling novices the capability to handle process models autonomously due to the difficulty of using the required modeling software, languages, and the models themselves [NP12]. Consequently, it may happen that some individuals are not able to contribute in practice despite having the theoretic possibility to do so [SN09]. This problem may be overcome by teaching Social BPM participants the skills that are needed for successful participation.

As a short-term measure, it has been suggested that real-time collaborative process modeling can serve as a means for training employees that encourages the exchange of information about the depicted domain and BPM expertise [RMH13]. In this manner, more experienced individuals can help novice modelers to increase their level of expertise, and thus enable them to make a broader range of contributions with higher quality in the future. One promising concept that embodies this idea is Social BPM labs [CCL+13, ACUV16]. In such workshops, participants are familiarized with BPM by using social software and other tools for communication to collaboratively model the processes of a fictional (or real) organization over the course of a few days. While the general work organization is planned and controlled in a bottom-up fashion, top-down guidance is provided (e. g., by moderators, method experts, and quality managers) to ensure that the learning goals of the lab can be met. Ultimately, this may not only help participants to develop new skills, but can also serve as a means for promoting Social BPM and increasing the motivation of process stakeholders to participate.

Focusing on the long term, teaching BPM skills at universities through comprehensive curricula is another promising measure for BPM education that can firmly embed this knowledge within the mindset of future process modelers, and thus ultimately, organizational culture [BICS07b]. However, academic BPM education is faced with its own challenges that need to be overcome first, such as creating appropriate teaching resources, and properly embedding BPM courses within the larger contexts of different degree programs [BCC+10]. For instance, Pflanzl et al. have recently presented a concept for an introductory lecture for IS curricula which suggests the use of a process modeling case study to teach basic BPM skills to students directly in the first semester [PBSV15a].

Additionally, education in a university setting has the evident drawback of being available only to a subset of the population. Thus, its primary purpose can be seen as creating BPM leaders who take responsibility for BPM, provide guidance to other participants while respecting the principles of social software, and promote Social BPM to further stakeholders.

The necessity of further measures for BPM training and education has been recognized and discussed in several publications, [BICS07b, IGRR09, Trk10]. However, to date it is a topic being addressed mostly by practitioners, whereas academics are focusing their efforts on open problems concerning methods and technology [IGRR09]. Due to the importance of this challenge for Social BPM, researchers should invest additional effort in examining which skills are required for which exact contribution in the BPM life cycle, and how it is possible to teach these skills most appropriately.

4.3 Gamification Potentials

The three main outcomes of gamification mentioned most consistently across the relevant literature are improving task performance, increasing motivation, and facilitating learning experiences [HKS14, SF15b]. As it can easily be seen, these benefits directly correspond to the three challenges of Social BPM discussed in detail in the previous section. Therefore, the biggest impact of applying gamification to Social BPM can be achieved by designing a gamified business process modeling tool that leverages the potential impacts of the former to solve the corresponding problems of the latter. This explains the focus of the detailed discussion in the previous section and here on these particular issues. The remainder of this section gives an overview of how gamification can be used to achieve higher levels of model quality in Section 4.3.1, to increase the motivation of potential participants to model in Section 4.3.2, and support BPM training and education in Section 4.3.3.

4.3.1 Improving Model Quality

One of the most comprehensive definitions of the term “game” (see Section 3.4.4) was proposed by Juul and reads as follows: “A game is a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable” [Juu11, p. 36]. Based on this understanding, business process modeling can also be understood as a game with the following characteristics (cf. Section 2.3):

- **Rule-based formal system**: The rules of process modeling are given by the process modeling language, in particular its syntax and primary notation. These rules can be interpreted as the mechanics of process modeling that in turn give rise to its dynamics [HLZ04].
- **Variable and quantifiable outcome**: For any given business process, the number of process models that can be created is theoretically infinite. Even under the assumption of fixed semantics, unlimited model variations can be created through secondary notation. By applying quality metrics, objective quantitative indicators can be calculated for any process model.
- **Outcomes are assigned different values**: Quality metrics are inherently connected to a valuation of process models. Thus, given two different models representing the same real-world excerpt, a statement about which of the models is better can be made. This makes it possible to select the best alternative out of a set of equivalent choices.
- **Player exerts effort**: Models do not “fall from the sky”, but are created by modelers. The more effort an individual expends, the higher the value of the outcome can be.
- **Player feels attached to outcome**: Exerting effort can make modelers emotionally attached to the outcome of the modeling activity. This may lead to varying emotional responses (e. g., happiness, frustration) depending on the value that is attached to the modeling outcome.
- **Consequences are optional and negotiable**: The created process model may ultimately change how an organization conducts its business, but this does not necessarily have to be the case. Furthermore, any consequences for the participating modelers should be optional and subject to

negotiation. In particular, there should be no imminent threat of negative consequences, as this contradicts the principles of Social BPM.

An important characteristic of good games is that they are highly successful in eliciting maximum task performance from their players. Consequently, "in a good computer or video game [the player is] always playing on the very edge of [his] skill level, always on the brink of falling off" [McG11, p. 24]. This allows players, among other things, to complete the most difficult levels, defeat the strongest monsters, find even the most well-hidden treasures, and explore every last corner of vast game worlds. To achieve this, game designers rely on a large variety of different principles, patterns, mechanics, and heuristics, such as providing clear goals, giving immediate performance feedback, matching the levels of player skill and challenge, and supporting skill acquisition from the beginning until the end of play (see Section 3.4.3). The fact that business process modeling is an activity that can already be interpreted as a game suggests that the performance-maximizing function of games can also be transferred to this context. To that extent, the task of creating a high-quality process model must be subdivided into multiple smaller tasks that correspond to individual quality metrics. Furthermore, modelers must be provided with appropriate quality feedback while modeling that allows them to operationalize the aforementioned metrics. Under the assumption that the set of available metrics addresses all aspects of what makes a process model "good", this may allow the participants of Social BPM to collaboratively work towards maintaining high model quality.

4.3.2 Increasing Motivation to Model

Playing a (good) game is a highly enjoyable experience (see Section 3.4.4) that can evoke various emotional responses such as sensation, fantasy, discovery, and expression [HLZ04]. Furthermore, it can satisfy the innate human need for autonomy, competence, and relatedness [RRP06] and trigger flow experiences during which players become so embedded in the activity of playing that they lose track of time [SW05]. Due to these characteristics, McGonigal argues that games "are fulfilling genuine human needs that the real world is currently unable to satisfy" [McG11, p. 4]. Irrespective of the extent to which that statement is correct, it is a fact that nowadays a large fraction of people across the globe play (digital) games and invest significant time and money for doing so [36, 37, Ent16]. Consequently, there is no room for doubt about the motivational capabilities of computer and video games.

In the context of BPM, the motivational situation is much more uncertain. Focusing on method and technology experts for whom activities related to the BPM life cycle are explicit parts of their job profiles, it can be presumed that they are at least extrinsically motivated (e. g., through their salary or the fear of being reprimanded) to also participate in Social BPM. Looking at other process stakeholders, this might not necessarily be the case. Theoretically, it can be assumed that such employees should be intrinsically motivated to contribute to Social BPM, as this gives them the opportunity to change their own work life for the better. However, since this is not part of their core work activities and it can take considerable time for the consequences of participation to manifest—if at all—many stakeholders may choose to abstain from investing into Social BPM. As the underlying principles of social software mandate that participation should occur on a voluntary basis, this poses a considerable problem. Furthermore, some authors note a general resistance of employees against any kind of BPM initiative and the difficulty of securing “user buy-in” [BKR03, Ros06a]. Overall, this raises doubts about the feasibility of Social BPM without the implementation of additional systems of motivation.

To bridge this motivational gap, gamification may be used. For instance, the gamification “blueprint” consisting of points, badges, and leaderboards represents a source of extrinsic motivation that can move users towards a certain behavior by providing them with external goals such as “getting points”, “unlocking badges”, and “beating others”. Furthermore, gamification can also address the inherent human needs for autonomy, competence, and relatedness to support intrinsic motivation and facilitate the internalization of external motives. In various studies on work motivation, this has been found to yield the outcomes of “(1) persistence and maintained behavior change; (2) effective performance, particularly on tasks requiring creativity, cognitive flexibility, and conceptual understanding; (3) job satisfaction; (4) positive work-related attitudes; (5) organizational citizenship behaviors; and (6) psychological adjustment and well-being” [GD05, p. 337]. Consequently, addressing intrinsic motivation is an important precondition for the success of Social BPM on both the individual and the organizational scale.

4.3.3 Improving Training and Education

A perspective on games that can often be found in game design literature views them as tools for learning. For instance, Koster suggests that "[when] you're playing a game, it exercises your brain" [Kos05, p. 39], and that a good game "teaches everything it has to offer before the player stops playing" [Kos05, p. 46]. Empirical studies have found that games can not only teach new knowledge, but also motor, cognitive, social, and emotional skills [CBM+12]. This is argued to not only be an effect of games, but one of the main sources of enjoyment that comes from playing them [Kos05, SW05, McG11], or in other words, "[fun] is just another word for learning" [Kos05, p. 46]. This position has also been adopted by gamification researchers, which is reflected in a large number of gamified systems that have been developed for education and teaching [NZT+14, SF15b]. Similarly, many of the most popular gamification examples from practice are also related to self-improvement, e. g., by becoming a better runner or learning a new language (see Section 3.2.1).

Carried over to Social BPM, gamification can serve as an approach through which the task of learning business process modeling can be converted into a gameful activity. For instance, following game design principles and heuristics for an effortless beginning of play (cf. Section 3.4.3), it should be possible to start modeling with a software without having to read a manual or documentation. To that extent, a series of modeling tutorials could be provided, of which each is designed to teach the user one new element of the utilized modeling language or function of the tool. These tutorials should be presented as a meaningful, coherent scenario (e. g., a sample enterprise) so that the tasks to accomplish feel authentic. Furthermore, they should be optional so that they can be skipped by more experienced modelers. Further instructional impact can be expected from the combination of quality metrics with real-time quality feedback. By being able to observe the effects of their actions on model quality, users can learn about the properties that a good model should possess, how good model quality can be achieved, and which relationships between individual metrics exist. Overall, the most significant impact of integrating gamified teaching facilities into a business process modeling tool is that this makes the acquisition of modeling skills a self-determined activity, thereby potentially increasing motivation and allowing additional educational measures (e. g., workshops and courses) to focus on other core elements of BPM.

4.4 Related Work

Efforts within the BPM discipline to conduct research at the intersection with gamification are still at a very early stage, and generally speaking, no sophisticated gamification concepts or tools incorporating game design elements as an integral part rather than an afterthought can be found yet. However, many authors have recognized the potential of gamification for BPM—in particular Social BPM—and calls as well as declarations of intent to conduct such research have recently become more frequent [Bra13, AEHO14, SAK15].

Some of the research that has been conducted to date is purely conceptual in nature. For instance, Erol et al. propose using "honour points" as a simple reward mechanism for activities conducted by Social BPM contributors to ensure their voluntary participation [EGH^{+}10]. Upon collecting enough points, modelers can exchange them for more tangible rewards such as acknowledgments, monetary reimbursement, or the certification of having fulfilled certain organizational requirements.

Another proposal by Brambilla introduces a platform for personal process management, a field related to Social BPM focusing on the private processes of individuals, and suggests using gamification as a "first class citizen" in the approach [Bra13]. Specifically, the author mentions the use of points and badges to increase user engagement. This work is advanced in [BRVB15], but despite the repeated claim to include gamification as a "first class citizen", no additional details about the proposed gamification design can be found.

Other authors describe concrete implementations of shallow gamification concepts which include game design elements as ancillary components of secondary importance rather than integral parts. Firstly, Awad et al. present *ISEAsy*, a gamified tool for participative end-user process modeling. ISEAsy is based on the *ISEA* method, which defines its own BPM life cycle and modeling language [ADL13]. While the authors elaborate on the theoretical impact of gamification on the motivation of individuals and teams, the actual gamification design is limited to an experience points and level system and its discussion remains superficial. In particular, no detailed information is provided about the activities for which points are rewarded, and the benefits that gaining a level provides. Finally, the lack of a proper evaluation does not allow for any conclusions regarding the effectiveness of the implemented approach.

Secondly, Smedt et al. suggest using gamification as a means for introducing modeling novices to the declarative process modeling language *Declare* and its more difficult aspects in a game-like fashion [DDSV16]. To that extent, they describe the implementation of a guessing game in which players must predict how the state of a process instance changes over time as its activities are executed. The authors further mention that the number of guesses can be used to derive a points score, but do not give any additional details.

Lastly, in discussing the success factors of collaborative business process modeling, Rittgen determines motivation for modeling as an important key determinant [Rit10b]. The author also discusses the underlying idea of gamification without using the term, i. e., "people gladly do a tough job if it comes in the disguise of a game. Think of PC gamers who spend days and nights without monetary reward just to reach the next level" [Rit10b, p. 29]. Using this metaphor as a foundation, the author describes a competitive modeling game in which participants score the models created by others and the "best" model is nominated as a winner. Such a competition is argued to serve as a source of extrinsic motivation. However, no details about the exact scoring mechanism are provided.

A more sophisticated approach is presented by Hoppenbrouwers and Schotten, who interpret business process modeling as a game consisting of a minimal set of modeling goals, immediate feedback using graphics and sound, a score, time pressure resulting from a reduction of the score over time, and the hope that modelers are motivated to play [HS09]. The score system proposed by the authors rewards modelers with 100 points for each process activity, 100 points for each connecting arc, and 10 points for the specification of input and output objects. The authors further describe a software prototype implementing such a game that can be seen as a gamified process modeling tool. Upon playing the game, the tool is capable of delivering a formal process model, for instance as a BPMN diagram. Compared to the previous approaches, the proposal by Hoppenbrouwers and Schotten is built on a game design perspective as its foundation rather than only considering it as an afterthought. Thus, it can be regarded as the most sophisticated realization of gamified business process modeling existing to date. However, it should be pointed out that the scoring scheme presented by the authors is drastically limited in the sense that it rewards quantity of work rather than quality.

Focusing on the distributed execution of business processes in crowdsourced settings rather than modeling, Scekic et al. define *RMod*, a model for the specification of reward and incentive mechanisms based on rules and events [STD12]. The system proposed by the authors allows distributing rewards based on conditions such as performing better than others or surpassing a particular lower bound. Further mention is made of the possibility to use the system as a foundation of a badge system. Overall, RMod is situated at a more general level than the proposals described so far and can be used for the realization of reward systems in any gamified software application.

Further related work can be found in the area of serious games, whose main difference from gamification lies in the creation of full-fledged games instead of non-game applications with individual game elements (see Section 3.3). For instance, Brown and various collaborators discuss the application of virtual worlds to process modeling, thereby effectively rendering the latter a game-like activity set in a 3D environment [BRW11]. This approach is proposed as a tool for facilitating modeling in distributed, collaborative scenarios involving not only experts, but also novice users [Bro10, GBR12]. Another example is *Innov8 2.0*, a serious game developed by IBM designed to educate players about the importance and benefits of BPM for organizations [BL13]. To that extent, players assume the role of a consultant and are tasked with re-engineering specific processes to improve their effectiveness and efficiency [SJ09].

Overall, researchers have recognized the potential of gamification to motivate relevant process stakeholders to participate in business process modeling and other BPM activities [HS09, EGH+10, Rit10b, ADL13], and to educate and train [SJ09, BL13, DDSV16]. However, many of the ideas that have been presented to date remain superficial in the extent of the proposed designs—for instance by focusing on work quantity rather than quality. Furthermore, experience reports and empirical studies describing the impacts of gamification on, e. g., motivation and the quality of process modeling outcomes have yet to be published. Thus, any suggestions about the benefits of gamification in the context of BPM remain conjecture and speculation despite the support of theories such as those described in Section 3.5. Ultimately, this leaves a considerable research gap and an opportunity for proposing more sophisticated gamification designs for BPM—in particular process modeling—as well as a rigorous evaluation of the impacts of gamification using appropriate instruments.

4.5 Outlook

In the introduction of this thesis, the following research question was presented (see Section 1.2): *how can gamification be used to support business process modeling*? Here, the question was framed against the backdrop Social BPM and its challenges, and three particular potentials of gamification were identified, namely improving model quality, increasing the motivation of Social BPM participants to model, and improving training and education. As demonstrated in the previous subsection, very little related work exists at the intersection of BPM and gamification, and thus the research presented here aims to make a contribution towards closing this gap. To that extent, its main goal lies in the conceptualization, implementation, and evaluation of an IT artifact that allows leveraging the potentials of gamification for Social BPM. This artifact was realized as a gamification module for the HBM, a commercial business process modeling software that is described in Chapter 5. Afterwards, two perspectives on the created artifact are presented, namely a conceptual view focusing on its design in Chapter 6, and a technical view addressing its implementation in Chapter 7. Finally, the gamification module is evaluated in Chapter 8 with the goal of demonstrating the extent to which the artifact contributes to overcoming the challenges of Social BPM.

5 Context Description

In this chapter, the organizational and technological context of this research project is described. To that extent, Section 5.1 first provides an outline of the most important stakeholder roles that are commonly found in game design projects and adapts them to the context of this particular gamification endeavor. Section 5.2 then describes the fundamentals of the Horus method, a modeling method (cf. Section 2.3) for the holistic description of the business processes of an organization from multiple perspectives. Finally, Section 5.3 gives an overview of the Horus Business Modeler (HBM), a software tool implementing the Horus method into which gamification functionality is integrated.

5.1 Stakeholders

Creating a game can be arbitrarily complex, ranging anywhere from small one-man projects such as VVVVVV (Terry Cavanagh, 2010) to perennial, multi-million development endeavors involving several hundreds of people such as Grand Theft Auto V (Rockstar Games, 2013). Independent of the size of the project, people involved in the creation of a game may assume several roles, some of the most important being the following [Bat04]:

- **Publisher**. The publisher finances the development of a game and thus has the final say about whether a proposed game idea will be developed or not. Furthermore, the publisher may impose certain requirements on the project, such as the date until which the game must be shipped, the type of game that should be developed or the franchise to which the game should belong, or the profit model for which the game must be designed. The publisher is usually responsible for marketing and distributing a game and owns it once it has been finished.

- **External producer**. The external producer is a member of the publishing company and is responsible for the delivery of a finished game satisfying the given requirements on time and on budget. He is responsible for selecting an (internal or external) developer and ensuring that communication with the latter is as frictionless as possible. If required, the external producer may also need to act as the advocate of the game developer towards the producer to ensure fair treatment. Lastly, this role may also be directly involved in the design of a game, especially with regards to finding a trade-off between the features of the game and its development budget.

- **Internal producer**. The internal producer has the role of a project manager and is tasked with managing the development team and the development process of the game. Consequently, he is also responsible for assembling a suitable team at the beginning of the project. During the development process, the internal producer must evaluate any changes that are made to the concept of a game with regards to its potential impacts on achieving the goals of the project. Ultimately, the internal producer also decides at which point the game is finally ready to be shipped to the customers.

- **Designers**. There are different types of designers, who, generally speaking, conceive the essential rules and structure of a game. The *game designer* is often the visionary and develops the actual concept of a game and plans its mechanics and dynamics. Thus, he must ensure that the final product is actually fun to play for the intended players. Next, the *writer* has the task of creating the story of the game and writing additional textual contents such as the narrative, character dialogue, or the instruction manual. Lastly, there may also be *level designers*, who, as the name of the role indicates, are responsible for designing the individual levels of the game and the challenges they contain.

- **Programmers**. The programmers carry the burden of creating the technical implementation of the game design. Thus, they must also communicate with designers regarding the technical feasibility of various requirements and possible alternatives and limitations. Possible components of a game on which programmers may need to work include artificial intelligence, rendering, physics, and networking code.

- **Artists**. Besides source code, a large variety of different assets are part of what is ultimately shipped, including artwork, models, textures, backgrounds, animations, sounds effects, music, voice recordings, videos, and localizations. For all of these content types, the development team must include expert artists capable of creating the required assets. Besides internal employees, such experts may also be external parties hired for this explicit purpose.
- **Testers**. Testing is a task that is performed throughout the entire development process rather than just at its end. It is carried out by a number of testers aiming to answer questions such as “Is it fun?”, “Is it easy to use?”, “Does it makes sense?”, and “Does it work?”. To that extent, testers must create detailed reports allowing programmers to resolve potential issues. Testing is supervised by the *test lead* who manages the testing process and acts as an interface between the testers, designers, and developers of a game.

Most of these roles can equivalently be found in gamification projects and are also present in the context of the research project which has given rise to this thesis. Here, they are distributed among two primary stakeholders: The Horus software GmbH based in Ettlingen (Germany) which is a member of the PROMATIS group, and the Databases and Information Systems (DBIS) Group based in Münster (Germany) which is part of the Department of Information Systems at the University of Münster (Westfälische Wilhelms-Universität (WWU) Münster). Whereas the Horus software GmbH mainly assumes the roles of publisher and external producer and is thus the owner and main user of the delivered product, most of the remaining roles—in particular those related to the development of the gamification module—are represented by the DBIS Group. More specifically, whereas the primary supervisor of this thesis, Prof. Dr. Gottfried Vossen, can be seen as the internal producer, the author of this thesis acted as the main designer and programmer. Additionally, testing was carried out by employees of the Horus software GmbH, fellow doctoral candidates at the DBIS group as well as students having attended the lecture *Introduction to IS* from 2014 to 2017 (see Chapter 8).

5.2 Horus Method

The Horus method is a product of Horus software GmbH located in Ettlingen, Germany, and is documented in detail in the accompanying book authored by Schönthaler et al. [SVOK12]. It represents an integrated approach for modeling business processes and their organizational contexts that is based on XML nets, a specialization of Petri nets, as a core modeling language. Additional modeling languages are also incorporated for the specification of, e. g., business objects, organization structures, roles, and resources. Thus, the basic idea of the Horus method lies in modeling each perspective on the business processes of an enterprise using the most appropriate language and eventually connecting these views by joining elements of different model types with each other. This principle is illustrated in Figure 5.1, which depicts some of the most common perspectives considered by the Horus method and the relationships between them: procedures–roles (which activities are carried out by which roles?), procedures–resources (which resources are required for activity execution?), procedures-objects (which business objects are consumed/required and produced by activities?), roles–organization structure (which roles are assumed by which organization units?), and objects–organization structure (which organization units are responsible for which business objects?).

Being a modeling method in the sense of the term that was introduced in Section 2.3, the main purpose of the Horus method is to provide “guidance and assistance in the preparation of models” [SVOK12, p. 61] by defining guidelines for modeling projects that have proven to be effective in practice, thereby facilitating the application of Petri nets in the context of BPM. To that extent, it is subdivided into the four distinct phases shown in Figure 5.2 that are supported by accompanying *project management*, *quality assurance*, and *documentation* activities. For each phase except preparation Schönthaler et al. specify a comprehensive process (naturally also modeled as a Petri net) detailing the activities that it consists of, the order in which they are carried out, the inputs that are required, and outputs that are created [SVOK12].

Phase 0: Preparation. In this phase, the modeling project to be undertaken is defined and then initialized. Project definition consists of the specification of the project scope (i. e., which parts of the organization should be examined), the alloted time and budget, as well as project goals and their economic relevance.

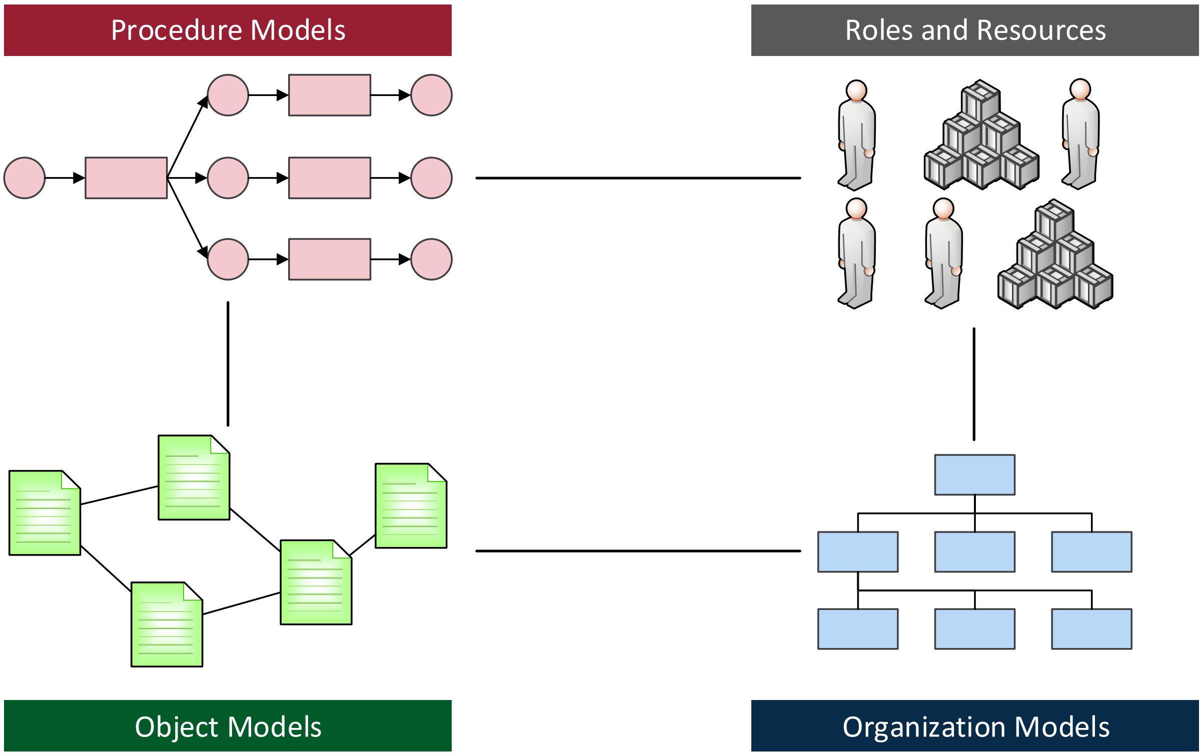


Figure 5.1: Main perspectives of the Horus method. Source: based on: [SVOK12, p. 28].

Phase 1: Strategy and architecture. The first phase of the Horus method has the main purpose of involving decision makers in the project, thereby ensuring their ongoing support of the modeling activities. It consists of an analysis of the environment and strategy of the target organization, as well as a description of its high-level architecture. To that extent, the following models must be created: 1. A *context model* describing the environment of the organization, 2. an *objectives model* outlining strategic business goals, 3. a *supply and services model* describing how the organization creates added value, 4. a *Strengths, Weaknesses, Opportunities, Threats (SWOT) model* analyzing internal and external factors that may benefit or harm the organization, 5. a *strategy model* defining strategies for reaching the previously outlined goals, 6. a *Key Performance Indicator (KPI) model* specifying metrics through which the attainment of strategic goals can be measured, 7. a *risk model* complementing the SWOT analysis, albeit on a more detailed level, 8. a *business process architecture model* defining the top-level process groups of the organization, 9. an *object*

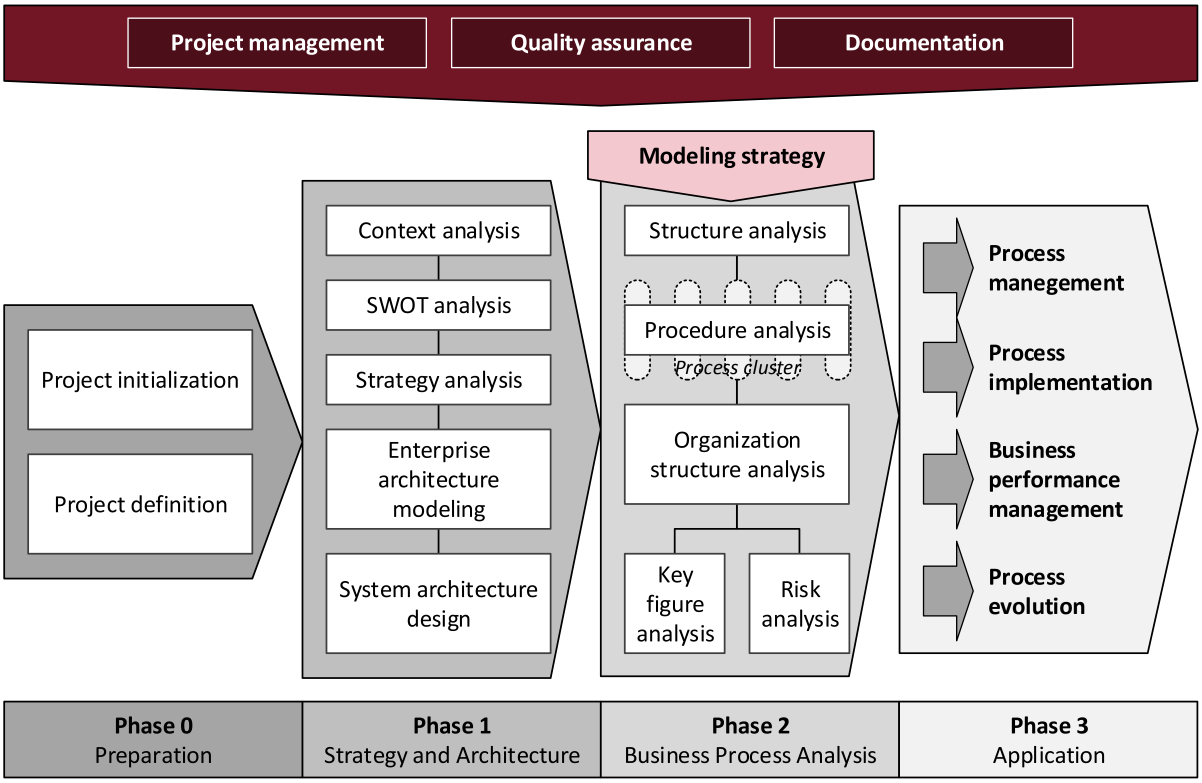


Figure 5.2: Phases of the Horus method. Source: based on: [SVOK12, p. 62].

model for business objects of strategic relevance, 10. a *business rules model* specifying global rules that must be respected across all business processes, 11. a *business units model* giving a high-level overview of strategic business units, and 12. a *system architecture model* describing hardware and software systems relevant for business processes. Following the principles of the Horus method, modelers must also specify the relationships between these models, for example by defining which KPIs can be used to measure the fulfillment of which strategic business objectives.

Phase 2: Business process analysis. Building on the results of the first phase, in the second step models are further refined and additional models created. This includes some of the most important aspects of business processes as illustrated in Figure 5.1. Of particular importance in this context is the refinement of process models, i. e., the specification of sub-processes on successively lower levels of abstraction based on the process architecture constructed in phase one. More specifically, the following models must be created or extended: 1. An *object model* providing detailed information about tangible (e. g., products, materials) and intangible (e. g., information, emails) business objects, including their attributes and relationships, 2. a *business rule model* specifying further constraints for business process execution as semi–formal tuples of events, conditions and actions, 3. a *procedure model* describing the actual business processes of the organization, 4. an *organization model* giving a detailed account of the organization units of the organization and their hierarchical structure, 5. a *resource model* providing information about human resources, machines, and other kinds of equipment required for process execution, 6. a *KPI model* with a definition of more detailed metrics on the level of processes and organization units, and 7. a *risk model*, again with more detailed specifications on the process or organization-unit level.

Phase 3: Application. In the final step, the models created in the previous phases are put to use, thereby connecting the Horus method to more large-scale Business Process Management (BPM) endeavors. Concrete application scenarios discussed in [SVOK12] include process implementation (i. e., the realization of an information system based on requirements derived from process models), business performance management (i. e., governing a business based on KPIs), and process evolution (i. e., restructuring business processes and their environment to improve their performance). However, all of the purposes of process models mentioned in Section 2.2 present further opportunities for applying the results of the Horus method.

Another important aspect of the Horus method that exploits the advantages of Petri nets while clearly distinguishing it from comparable approaches is its emphasis on using *simulation* throughout all phases. To that extent, the method is strongly supported by the Horus Business Modeler (HBM) (see the following section), which not only allows enriching models with information that is required for simulation, but also provides actual simulation capabilities. Generally speaking, the underlying idea of simulation is to create multiple variants of a business process that may differ in their parameterization and structure, and then compare them regarding performance metrics such as added value, costs, time, and quality. This allows decision makers to select the best alternative for a business process in accordance with the strategic goals of an organization.

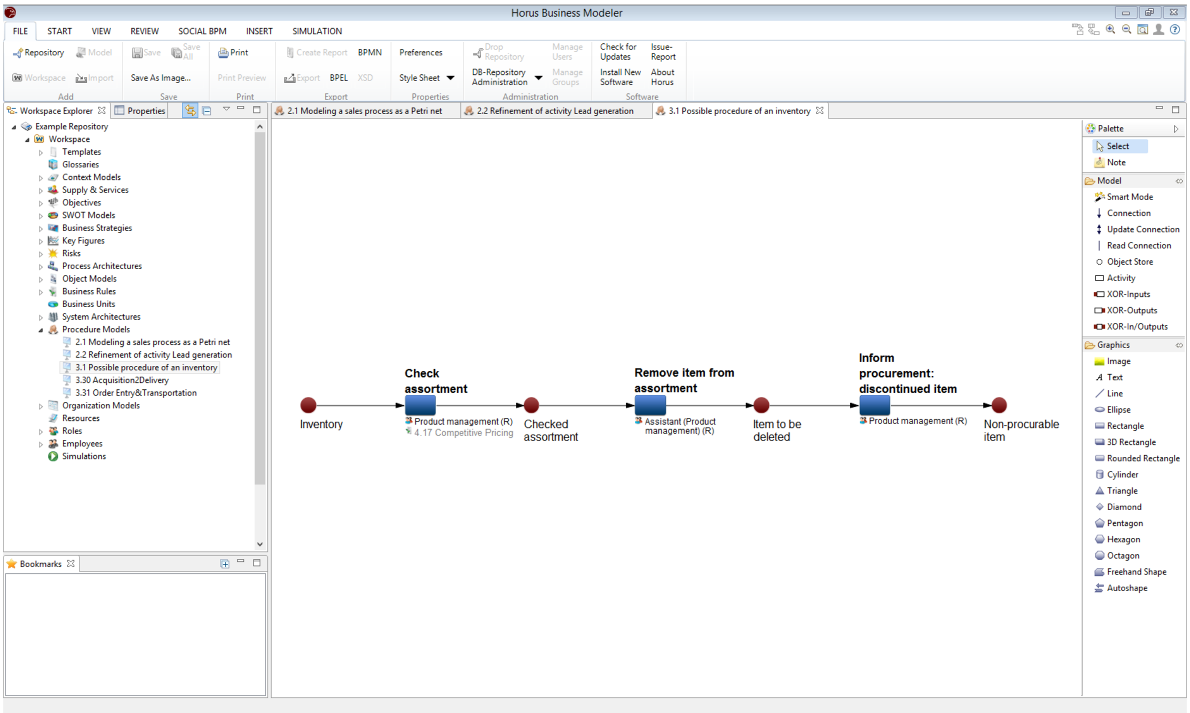


Figure 5.3: Interface of the Horus Business Modeler v2.6; process model in editor.

5.3 Horus Business Modeler

The Horus Business Modeler (HBM) is a desktop application[1] that provides tool support for BPM projects using the Horus method. It is based on the so-called Eclipse Rich Client Platform (RCP) that allows for the creation of rich software applications using the architecture, components, and toolkits of the Eclipse Integrated Development Environment (IDE). While development of the gamification module commenced for Version 2.5 of the tool, the final release was integrated into Version 2.6. As the most significant change between both versions lies in the introduction of a larger, ribbon-based menu bar in Version 2.6 as compared to a smaller, more traditional menu and tool bar in Version 2.5, all subsequent explanations will be based on the newest release and make no further distinction between the concrete versions of the HBM.

[1] Available at: http://www.horus.biz/de/download/. Last accessed: 2017-01-13.

The interface of the software is shown in Figure 5.3 and can broadly be subdivided into three different areas: the ribbon menu (top), the workspace explorer (left), and the editor (rest). Firstly, the *ribbon menu* works similarly to implementations provided in other tools such as the Microsoft Office suite and provides quick access to the most common functions categorized by application area. Secondly, the *workspace explorer* occupies the left-hand side of the remaining screen real estate and allows access to the actual modeling artifacts that are created while applying the Horus method. It is organized in a hierarchical, tree-like structure wherein different types of nodes can be distinguished:

- **Repository**. Situated at the top level, a repository represents a storage unit for model files that is physically and/or logically separate from other repositories. Three different types of repositories can be created: *local* repositories in which model files are locally stored on the hard drive of the user, *database* repositories in which model files are stored in a (local or remote) Relational Database Management System (RDBMS), and *server* repositories which build upon database repositories but provide further features and use Web services to communicate with clients.

- **Workspace**. Situated at the second level, a workspace represents a storage unit for model files that is logically separate from other workspaces. By creating workspaces, modelers can thus subdivide a repository into different modeling projects and sub-projects that are kept in the same physical storage. To that extent, workspaces can be nested with arbitrary depth to create workspace hierarchies as needed.

- **Model category**. Situated at the penultimate level, a model category represents a folder in which only models of the same type (e. g., process models, object models, or organization models) can be created. For any workspace, administrators may freely define a (potentially empty) subset of model categories that are enabled and visible.

- **Model**. Within the hierarchy of nodes in the workspace explorer, the actual models are always leaves, i. e., terminal nodes.

Double-clicking a model in the workspace explorer loads it into the *editor*, which is again subdivided into two areas: the modeling *canvas* occupying the

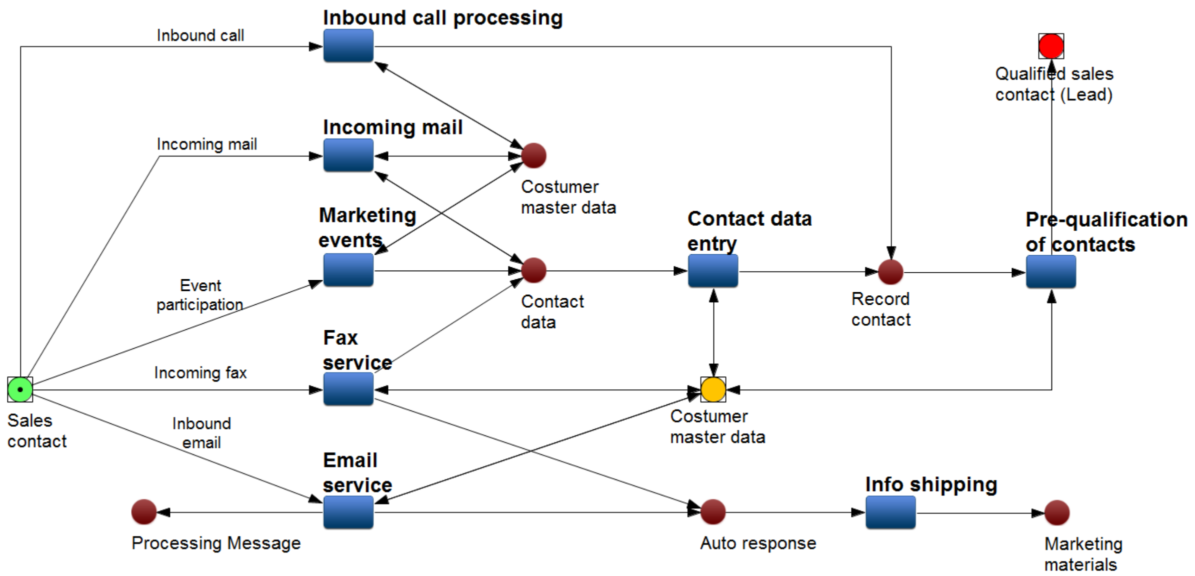


Figure 5.4: Petri-net process model.

majority of the remaining space and containing the model definition, and the *palette* of modeling elements on the right. As demonstrated by Figure 5.3, Horus realizes a Multiple Document Interface (MDI), meaning that multiple models can be loaded at the same time and are visualized using tabs. The concrete editor that is used when loading an existing model or creating a new one depends on the model category the former falls into:

- A **Petri net editor** is provided for business process models (called behavior models), process architecture models, context models, control models, and viewing simulation traces of process model simulation runs. The particular type of high-level Petri nets realized in the HBM are XML nets as described in Section 2.3. The Petri net editor provides a number of advanced modeling features, such as the top-down refinement of activities to model business processes on multiple levels of abstraction, simulating the flow of tokens, or arranging model elements automatically. A sample process model created using the Petri net editor is shown in Figure 5.4.

- An **Object model editor** allows the creation of object models using the Asset Oriented Modeling (AOM) approach to data modeling [DM02,

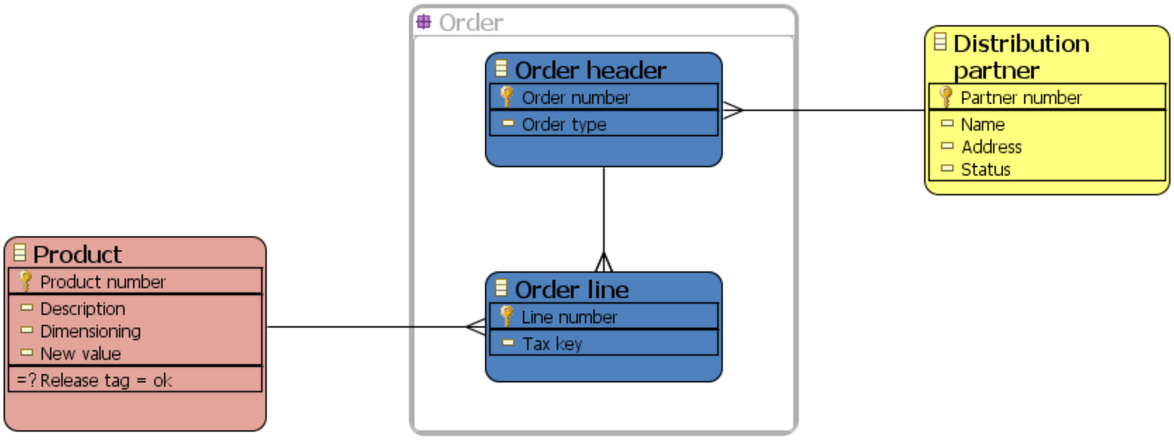


Figure 5.5: Object model.

Dau03], albeit with some deviations in the used terminology (e. g., "assets" are referred to as "objects" [SVOK12]). The most important concepts of this modeling language are objects, relationships between objects (including inheritance), and object attributes. AOM is based on the Higher Order Entity Relationship Model (HERM) [Tha00] and, as a direct result of this lineage, yields models with a structure that is very similar to XML schemas. Therefore, it suitably complements the use of XML nets for process modeling. From a graphical point of view, the models created using this editor are similar to UML class diagrams. A sample object model created using the object model editor is shown in Figure 5.5. Further information about AOM is not required in the context of this thesis, and thus the interested reader is referred to [DM02], [Dau03], and [SVOK12] for a more detailed discussion.

- An **Semantic Hierarchy Model (SHM) editor** enables the creation of a variety of models with a hierarchical structure based on the SHM approach to data modeling [SS77, BR84]. The most important concepts of this modeling approach are objects and relationships such as generalization, aggregation, and association. The SHM editor exists in various specializations (some with a reduced or modified set of relationships) and can be used to construct risk models, goal models, KPI models, SWOT models, strategy models, service models, skill models, business unit models, system architecture models, organizational structure models, and

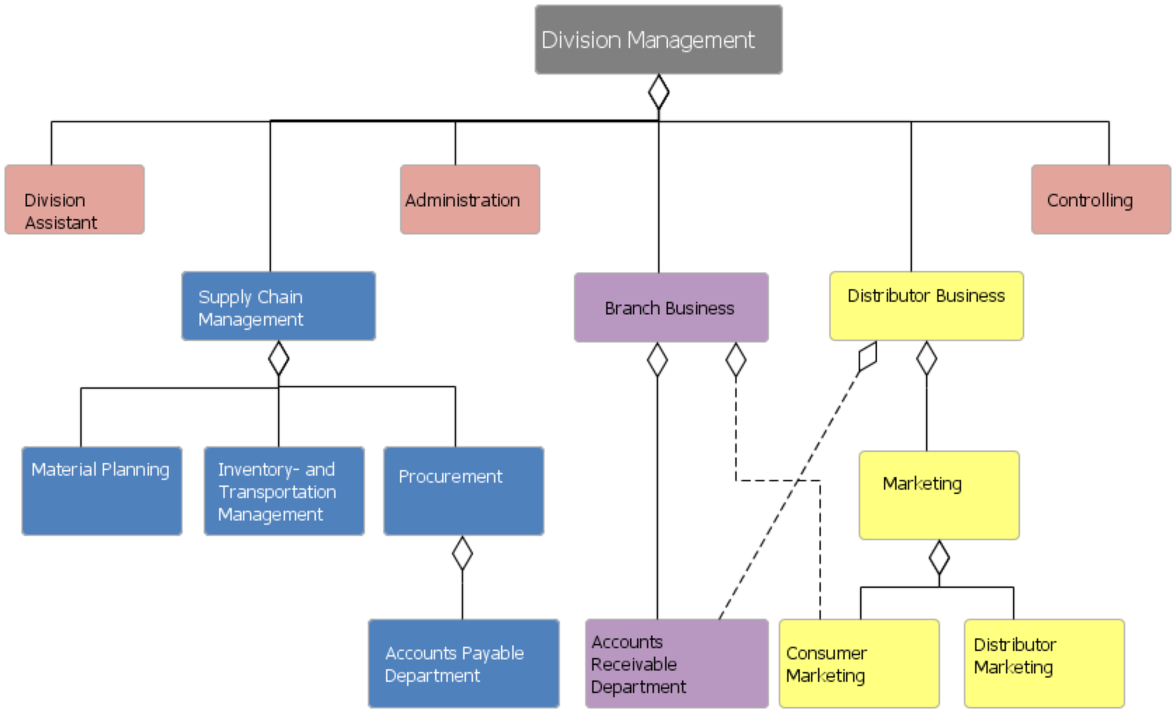


Figure 5.6: Organizational model.

resource models. A sample Semantic Hierarchy Model—specifically an organizational model—is shown in Figure 5.6.

- **Dialog-based editors** are provided for the creation of a variety of different models using text input fields, drop down lists, and other user interface elements rather than a visual modeling language. Consequently, the subdivision into a canvas and a palette as well as the possibility to edit multiple models at once do not apply for this type of editor. Dialog-based editors exist for the following model categories: glossaries, roles, employees, and business rules.
- A **Template editor** is provided for the creation of visual templates that can be used as the starting point of other models. This allows defining a uniform visual style for models that is consistently applied throughout a workspace.
- A **Simulation editor** serves the purpose of creating simulation runs of process models and process architectures.

- A **BPMN editor** is provided that can be used to create BPMN process models and choreographies as an alternative to the Petri net editor.

In most of the visual editors mentioned above, more detailed information can be provided about individual model elements by double-clicking them to access their *property dialogs* which are similar to dialog-based editors, but are used at a lower level of abstraction. Depending on the type of the selected model element, the property dialog is filled with different *property pages* corresponding to business properties that are relevant for the former, e. g., capacities for places in a Petri net, and execution times for transitions. Overall, two types of properties (and thus of property pages) can be distinguished. Firstly, *general properties* allow incorporating various pieces of domain knowledge into a model element, thereby greatly extending and clarifying the semantics of a model. For instance, the most important property page for Petri net places allows users to specify their capacity, the number of tokens contained and their fetch strategy, the storage cost for tokens, and how their quality changes over time. Secondly, *reference properties* allow specifying relationships between the elements of different models, which enables modelers to connect the various perspectives on business processes that the Horus method considers. For instance, as illustrated in Figure 5.1, in Petri nets it is possible to define which roles are responsible for the execution of which activities and which types of objects are consumed and produced by these activities.

6 Design

Once a gamification project has been initiated, its objectives have been clarified, and its context and target users have been analyzed, gamification designers must carry out the task of specifying which properties the gamified solution should have [MWHA17]. This bears strong similarities to game development, where a *concept document* detailing aspects of the proposed game such as genre, gameplay, story, and target audience must also be created [Bat04]. The main purpose of this chapter is thus to describe a concept for the integration of gamification into the context outlined in Chapter 5 for the user groups identified in Chapter 4 with the main purpose of achieving the goals presented in Section 4.3. After discussing the design concept in Section 6.1, Section 6.2 loosely adapts the concept of *design lenses* to analyze the gamification design from various, goal-oriented perspectives.

6.1 Gamification Concept

As Morschheuser et al. point out, many gamification professionals consider designing a gamified solution a highly creative process that consists of a brainstorming phase focused on the collection of as many ideas as possible, and a consolidation phase in which ideas are selected and then formalized as a design concept [MWHA17]. For this purpose, the following tools were used:

1. **Computer and video games**. With actual games being the main source of inspiration for gamification and its related disciplines, referring to (commercial) computer and video games for ideas is a natural approach and was also done here. Examples are presented in the following subsections and further details on the referenced games can be found in the ludography at the end of this thesis.

2. **Game design elements**. Another important tenet of gamification is the assumption that games and their creation processes can be subdivided into clearly identifiable components that can be adapted to other contexts. Such game design elements were also used in the creation of this design concept and were discussed in length in Section 3.4. Besides game design literature, an important source of ideas in this context were also gamification literature reviews (e. g., [HKS14], [TLB14], and [NZT$^+$14]) and proposed taxonomies (e. g., [RB13a]).

3. **Gamification examples**. Further inspiration was derived from successful gamification examples as presented in Section 3.2. In particular, the concept described in the following builds on the gamification "trinity" of Points, Badges, Leaderboards, which, despite the considerable criticism it has received, many gamification studies associate with positive (or at least mixed) outcomes.

4. **Social BPM examples**. Some parts of this concept are more strongly related to Social BPM than to gamification, i. e., they address aspects such as self-organized, collaborative process modeling in heterogeneous teams. Despite that, they are still firmly integrated with the core gamification functionality and thus presented here.

For each of the features discussed in the following, the most direct source of inspiration is indicated as faithfully as possible. An overview of additional tools that may be used for ideation along with suggested references can be found in [MWHA17].

In the interest of linguistic simplicity, the design concept is described from the perspective of an already existing de-facto implementation of its features as part of the HBM. The entirety of the concept is modeled in Figure 6.1 as a high-level UML class diagram using only inheritance and general dependency relationships while omitting multiplicities. In this visualization, blue classes correspond to activities within the HBM, red elements to rewards users can earn, and green classes to further gamification components, of which dark-green classes are user-facing visualizations. In the following subsections, elements contained in the design concept are grouped according to their primary function and discussed in detail. Specifically, Section 6.1.1 is focused on tasks users can perform, Section 6.1.2 on the rewards they can earn,

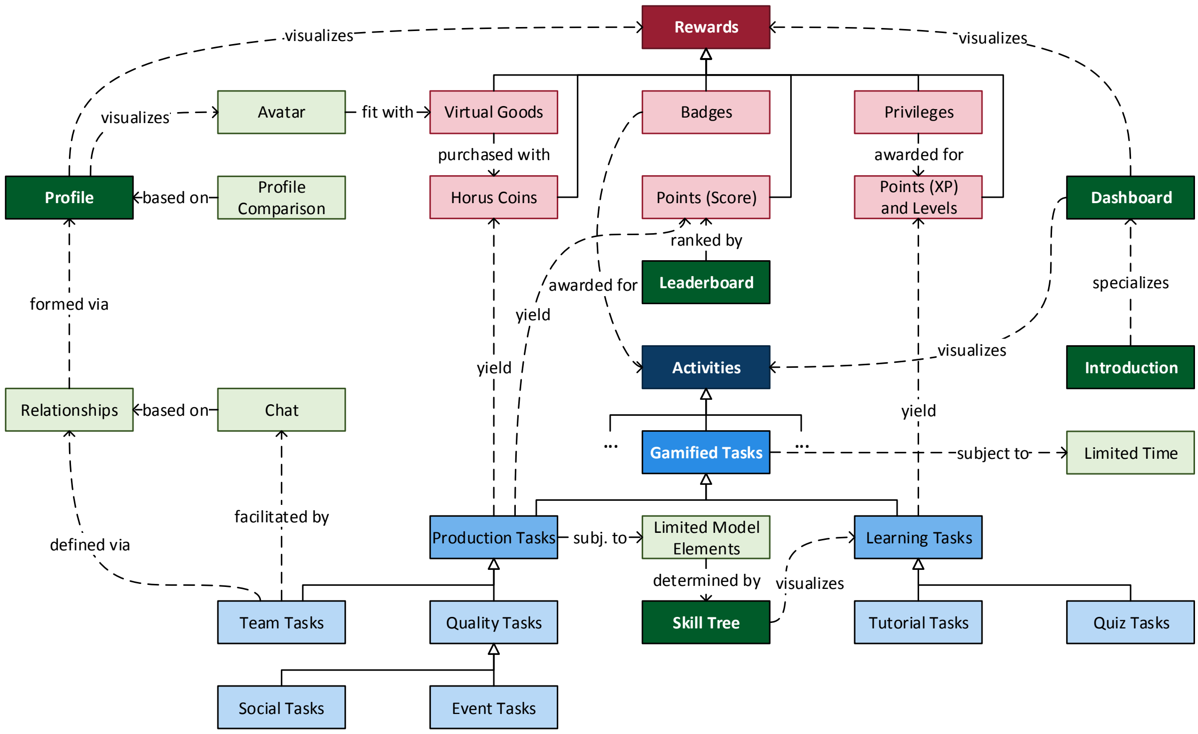


Figure 6.1: Gamification design concept: Overview.

Section 6.1.3 on the embodiment of players within the tool, Section 6.1.4 on possible sources for competition, Section 6.1.5 on facilities for user interaction, and Section 6.1.6 on the concept of limited resources.

It should be noted that for many of the elements presented in the following, distinctive bodies of research discussing them in detail can be identified in the game studies literature. However, as the purpose of this section is to describe a concrete design composition rather than to provide a comprehensive, annotated bibliography of game design elements, no attempt will be made to address them in their entirety. Thus, for further information on individual elements, the reader is referred to the game design literature referenced in Section 3.4.

6.1.1 Tasks

In some way, all gamified applications consist of certain gamified tasks that are highly domain-specific, e. g., gamified running for Nike+ or gamified language lessons for Duolingo. Drawing a parallel to games proper, these tasks assume a similar function as quests in RPGs, puzzles in puzzle games, or levels in First Person Shooters (FPSs). The activities that modelers may perform while using the Horus Business Modeler are very diverse and differ in the level of skill that is required for their execution. For instance, there are comparatively simple tasks that can be carried out by novice users without any modeling knowledge, such as reading description texts of models and model elements, as well as reading and understanding very simple models. Next, some tasks such as commenting on the contents of a model may require domain knowledge and rudimentary skills about the syntax and semantics of the underlying modeling language. Finally, doing extensive work on critical models requires users to posses even more skills, for instance knowing how to model control flow patterns such as alternatives or parallelism. This gamification concept envisions an extension of the set of possible tasks that are available in the HBM with *gamified tasks* that can further be subdivided into *production tasks* pertaining to the creation and modification of business process models using game design elements, and *learning tasks* focusing on gamified skill acquisition through work on ancillary models. Situated below these two elements in Figure 6.1, a total number of six concrete gamified tasks are described in the following paragraphs. Then, two additional paragraphs introduce further components that serve as enablers for the remaining tasks rather than as tasks themselves.

Quality tasks

Quality tasks relate to the quality of business process models and consist of a number of activities that are directly derived from the quality metrics that were presented in Section 2.7 as well as the characteristics of the Horus method and the Horus Business Modeler described in Chapter 5. Thus, every quality task represents a small, encapsulated work package whose final goal is the improvement of a single quality aspect of a model and that modelers can complete in a short amount of time. By performing multiple quality tasks in parallel or succession, users can execute more extensive modeling workflows that enable them to achieve an overall high model quality.

Generally speaking, the set of potential quality tasks is directly determined by all metrics presented in Section 2.7 for which an automated measurement procedure can be implemented into the HBM. However, to examine whether the proposed gamification concept can effectively be operationalized to maintain a high quality of business process models, it is not necessary for all metrics to be already implemented in an initial release. Thus, in the following it is suggested to start with a subset of metrics meeting the following requirements:

- Their significance for error probability, understanding, or some other factor with relevance for the use of process models has been demonstrated by empirical studies.
- They are easy to comprehend and their impacts are easy to understand and visualize. For instance, the fact that *edge crossings* and *node occlusion* have a negative impact on the readability of a process model can easily be recognized by simply looking at its visualization. This holds not only for modeling experts, but also for novice users. A negative example is the metric *cross-connectivity*, as part of its semantics are encapsulated by a complex, recursive valuing function.
- They are easy to implement and have a comparably low computational complexity, possibly through the use of appropriate heuristics. For instance, the metrics *size*, *density*, and *sources and sinks* have constant time complexity and can be determined through simple arithmetic operations based on set sizes. Conversely, metrics related to *textual labels* are more complex, as they are influenced by factors such as language, word stemming, synonyms, and homonyms.
- They are of particular significance to the Horus method. As mentioned in Section 5.1, one of the core principles of this modeling method is the description of business processes from multiple perspectives (e. g., procedures, objects, roles, and organization units), which are then linked together to enhance the expressiveness of the process model. Thus, metrics related to the *completeness* of such information are of special importance.

After comparing the quality metrics presented in Section 2.7 to these properties, one possible subset that can be specified is summarized in Table 6.1. Here, metrics have been assigned to three quality characteristics (thus foregoing the level of sub-characteristics proposed by Moody [Moo05]), namely *readability*, *complexity*, and *completeness*. For information on the bounds that were used for these metrics, the reader is referred to Section A.3 in the appendix. To arrive at quality scores for each characteristic and an entire model, the arithmetic mean of all relevant values can be used, unless weights such as those used in the 3QM framework are available. It should be noted that the selection of metrics presented here is only one of several valid choices, and other subsets or selection criteria could also be used. In fact, it can be argued that the underlying principle of providing modelers with gameful goals anchored to model quality is independent of the concrete metrics, and thus even an arbitrary subset could be employed.

Table 6.1: List of quality metrics in the Horus gamification concept.

Characteristic	Quality Metric	Optimization Goal
Readability	Edge Crossings	▼ Remove all edge crossings
Readability	Edge Bends	▼ Use bend points sparingly
Readability	Node Occlusion	▼ Remove node overlaps
Readability	Angular Resolution	▲ Maximize angles between leaving arcs
Readability	Consistent Flow 1	▲ Arrange elements from top to bottom
Readability	Consistent Flow 2	▲ Arrange elements from left to right
Readability	Orthogonality	▲ Lay model elements out on a grid
Complexity	Size	▼ Keep the size small, split large models up
Complexity	Diameter	▼ Minimize the value of this metric
Complexity	Density	▼ Minimize the value of this metric
Complexity	Connectivity Coefficient	▼ Minimize the value of this metric
Complexity	Avg. Connector Degree	▼ Minimize the value of this metric
Complexity	Max. Connector Degree	▼ Minimize the value of this metric
Complexity	Connector Mismatch	▼ For each split, model corresponding join
Complexity	Control Flow Complexity	▼ Limit the number of execution paths
Complexity	Cyclicity	▼ Try to avoid cycles
Complexity	Token Split	▼ Try to avoid AND and OR splits
Complexity	Sources and Sinks	▼ Use exactly one start and end element

Table 6.1 (contd.)

Characteristic	Quality Metric	Optimization Goal
Completeness	Names	▲ Provide names for all elements
Completeness	Short Names	▲ Provide short names for all elements
Completeness	Descriptions	▲ Provide descriptions for all elements
Completeness	Notes	▲ Provide notes for all elements
Completeness	Business Rules	▲ Provide business rules for all activities
Completeness	Documents	▲ Provide documents for all elements
Completeness	KPIs	▲ Provide KPIs for all activities
Completeness	Object Types	▲ Provide object types for all object stores
Completeness	Refinements	▲ Refine all activities with further details
Completeness	Resources	▲ Provide resources for all activities
Completeness	Risks	▲ Provide risks for all activities
Completeness	Roles	▲ Provide roles for all activities
Completeness	Services	▲ Provide services for all activities
Completeness	System Components	▲ Provide system comp. for all activities

For all quality tasks, it must be possible to compute the effects of any model changes on the quality metrics they are related to automatically and instantaneously. This allows providing modelers with real-time feedback about the effects of their actions on model quality and rewarding them based on the quantity and quality of their contributions. To that extent, the current quality of the model is continuously assessed on either the metric, the characteristic, or the model level. The result of this assessment is quickly presented at a prominent place within the editor view so that the modeler can effectively operationalize this information. If any changes carried out by the modeler result in a change of model quality, this is further highlighted to the user. Lastly, if the user decides to save the changes he made and the current quality of the model has increased since a save operation was last carried out, some reward will be granted to the former. Figure 6.2 illustrates how such feedback can be presented to the user by example of the quality task “remove edge crossings”.

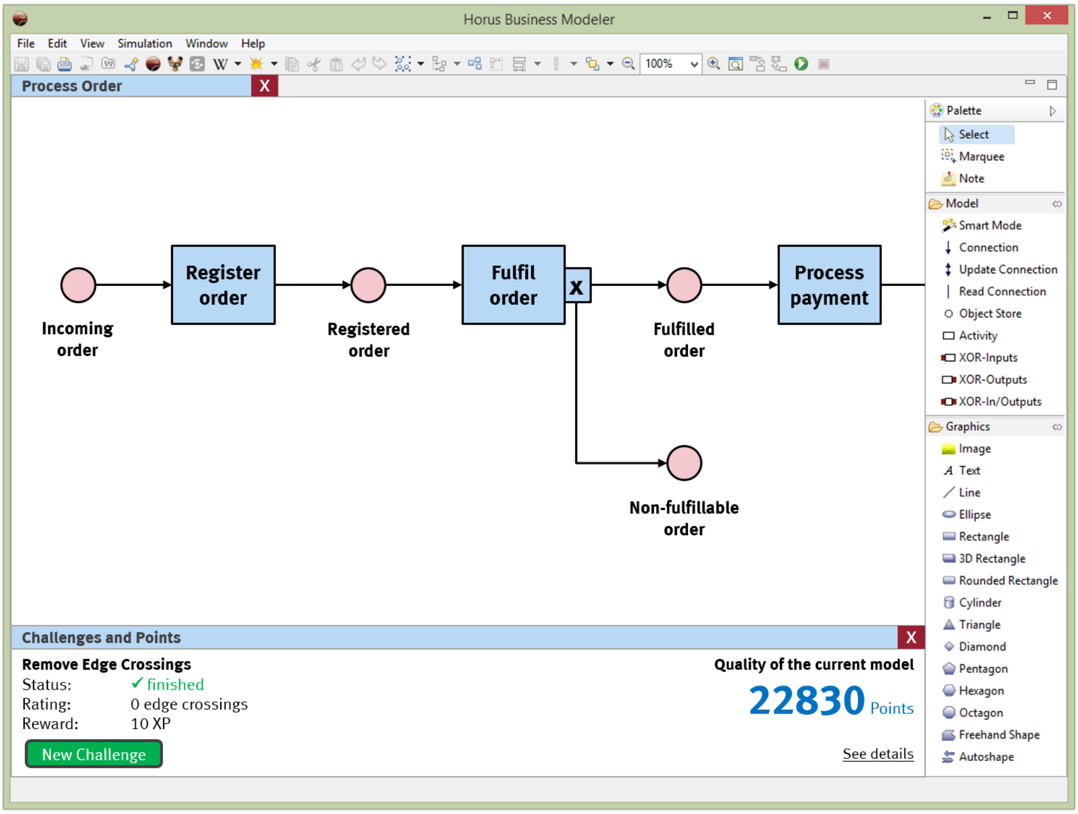


Figure 6.2: Gamification design concept: Quality task, feedback, and reward.

Here, the quality of the model is translated to a points score, although an indication as a percentage would be possible as well.

Social tasks

Social tasks are a specialization of quality tasks that focus on aspects of process models that are difficult or impossible to evaluate in a fully automated fashion and thus require human interpretation. As such, they for instance relate to the levels of *semantic quality* and *social quality* of the SEQUAL framework [KSJ06]. Thus, the main purpose of social tasks lies in evaluating the semantics of the statements expressed in a process model as well as the convergence in interpretations of these statements by different BPM stakeholders. Tasks that can be carried out in this context include the following:

- *Rate*: Users have the possibility to rate process models, for instance by awarding 1–5 stars or giving a “thumbs up” or “thumbs down” assessment (cf. [CM15]). This rating can either refer to the entire model as a whole, or to a certain aspect of its quality such as readability, completeness, and correctness. In addition, it is also possible to limit the scope of a rating to the contribution of a single modeler, i. e., the difference between two successive versions of the same model as shown in Figure 6.3. This allows giving special appreciation to users who have made noteworthy improvements.

- *Comment*: By writing comments, users can discuss particular aspects of a process model, for instance to point out errors or suggest improvements. It is possible to attach comments not only to a model as a whole, but also to parts of a model and individual model elements. Furthermore, it is possible to reply to previous comments to maintain continuous conversations. To a certain extent, commenting functionality is already provided by the HBM and thus merely needs to be extended and integrated into the gamification module.

- *Tag*: By adding tags to process models, elements within these models, groups of elements, or other digital artifacts, users collaboratively provide machine-readable details about their meaning. This information can then be used for various purposes, most notably improving the searchability of models and thus making it easier to filter relevant content in the light of information overload [HT85].

As the name already indicates, social tasks are strongly related to Social BPM, and thus further information can be found in pertinent literature (cf. [EGH+10, BDJ+11]). The distinctive feature in this context lies in the integration of social functionality with gamification. Furthermore, the main difference between social tasks and general quality tasks lies in the fact that the former exhibit strong limitations regarding the possibility to provide modelers with immediate feedback on the effects of their actions. For instance, ratings and comments on the contributions to a process model made by a particular modeler will not immediately be available but slowly accumulate over time, and thus rewards can only be granted in a delayed fashion. Therefore, special attention must be given to ensure that users are notified when other modelers carry out social tasks that affect them, for instance as shown in the list of recent events in Figure 6.8.

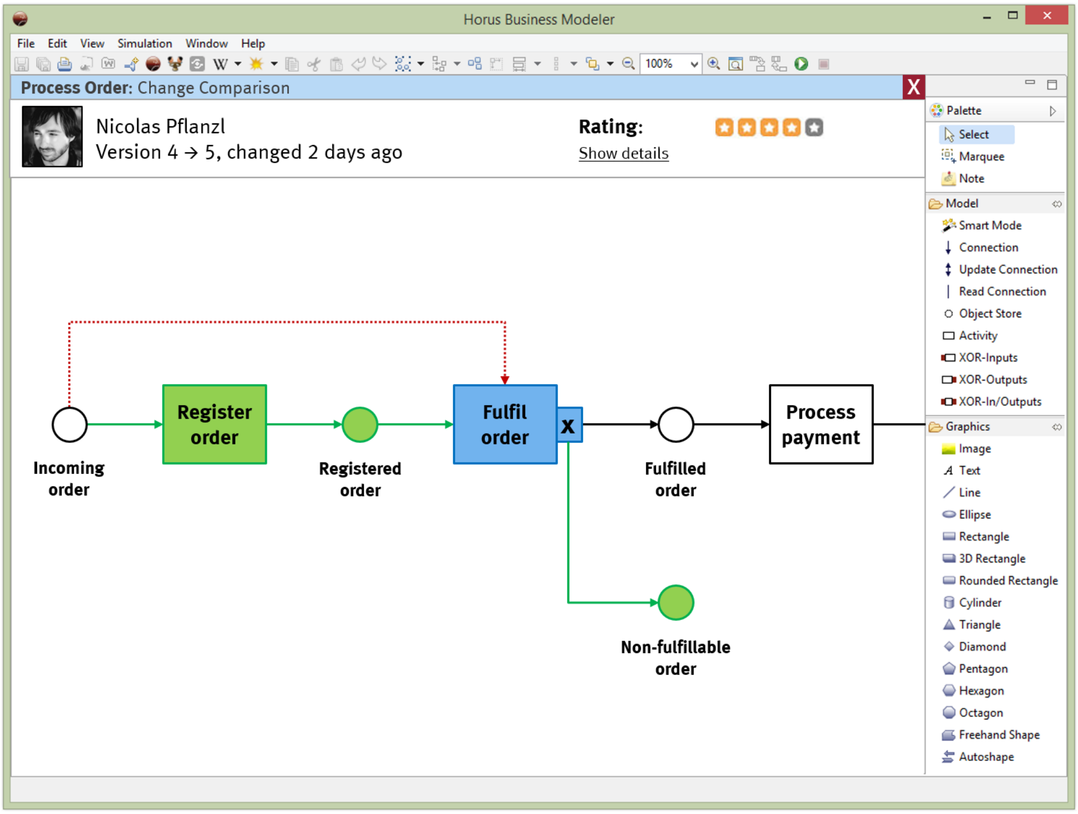


Figure 6.3: Gamification design concept: Model version comparison & rating.

Event tasks

An event task is a specialization of a quality task whose validity is restricted to a particular timespan and for which users can receive special rewards such as an increased number of points (see Section 6.1.2). This type of task draws inspiration from online RPGs in which similar events often take place. For instance, the publisher of Diablo III occasionally activated so-called "community buffs" during which players were able to earn more experience points from defeated monsters and the chance for superior pieces of equipment to drop was significantly higher. By defining event tasks, an organization can try to focus the effort invested by modelers on a particular type of activities for a limited period of time. The specification of current event tasks can for instance be a privilege that must be earned by users, carried out automatically by determining quality metrics exhibiting a low average value within a workspace or repository, or a secondary reward that can be bought with Horus Coins (see

Section 6.1.2). A list of event tasks that are currently active as well as their expiration dates is presented to users in a prominent place of the dashboard.

Team tasks

Compared to quality tasks, team tasks are characterized by a greater scope and longer time horizon. Furthermore, as the name indicates, they are not carried out by individual users, but by groups of multiple modelers with potentially heterogeneous sets of skills. Thus, team tasks share conceptual similarities with the core gameplay of many Massively Multiplayer Online Role-playing Games (MMORPGs). For instance, in World of Warcraft, many large-scale dungeons (so-called “raids”) are designed to present challenges to players that can only be overcome in groups of multiple characters with different abilities. Team tasks are always initiated by a single user who assumes the role of coordinator, moderator, and “game master”. Before the beginning of the task, the initiator defines its overarching goal, its due date, and a profile of the skills that the participants should possess as shown in Figure 6.4. Afterwards, other users can access a list of all proposed team tasks from the gamification dashboard and decide to join one if they have the required skills.

Once an appropriate team composition was found, the team task can be started. In the style of a simple to-do list, participants can define a list of the activities that must be carried out to conclude the team task successfully. To that extent, they may also refer to the repertoire of available quality tasks. The members of the team may coordinate their activities by making use of the communication facilities described in Section 6.1.5. Once the due date for the team task has been reached or the initiator has manually marked it as finished, the activity is terminated. Afterwards, all participants are rewarded based on the outcome of the task and the extent of their own contribution as shown in Figure 6.5. The size of the reward may depend on multiple factors such as the quantity and quality of the performed work as well as a “social” multiplier denoting the quality of collaboration. As these indicators may be difficult to compute automatically, the initiator may be required to assess them as objectively as possible. Lastly, participants have the opportunity to altruistically relay part of their own reward to other users if they wish to give them special recognition for their contribution to the final outcome.

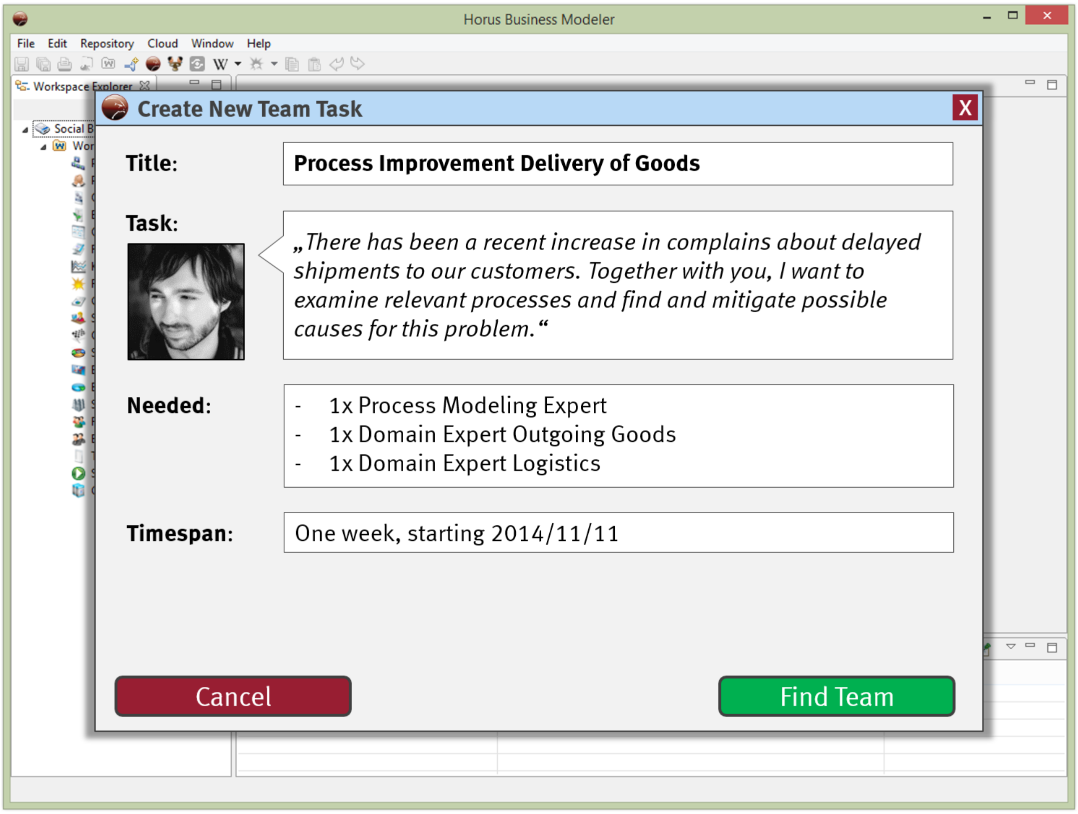


Figure 6.4: Gamification design concept: Team task creation.

Tutorial tasks

Tutorial tasks are an integral part of the learning facilities to be implemented into the HBM. As shown in Figure 6.6, they are structured into virtual scenarios with varying difficulty levels and set specific performance goals for modelers to achieve, such as performing a specific task within limited time. Upon completion of a tutorial, users may be rewarded with experience points or another virtual prize. Furthermore, tutorials are designed to teach particular skills to users and to become progressively more difficult. This shall give modeling novices the opportunity to learn business process modeling while actually modeling, and to keep track of their progress in doing so. While executing a particular tutorial, users are presented with step-by-step instructions and explanations teaching them the subject matter of the tutorial and gently nudging them towards the tutorial goals, such as depicted in Figure 6.7. Tutorial tasks are strongly inspired by their respective counterparts in video games. For instance, the strategy game Age of Empires II (Ensemble Studios,

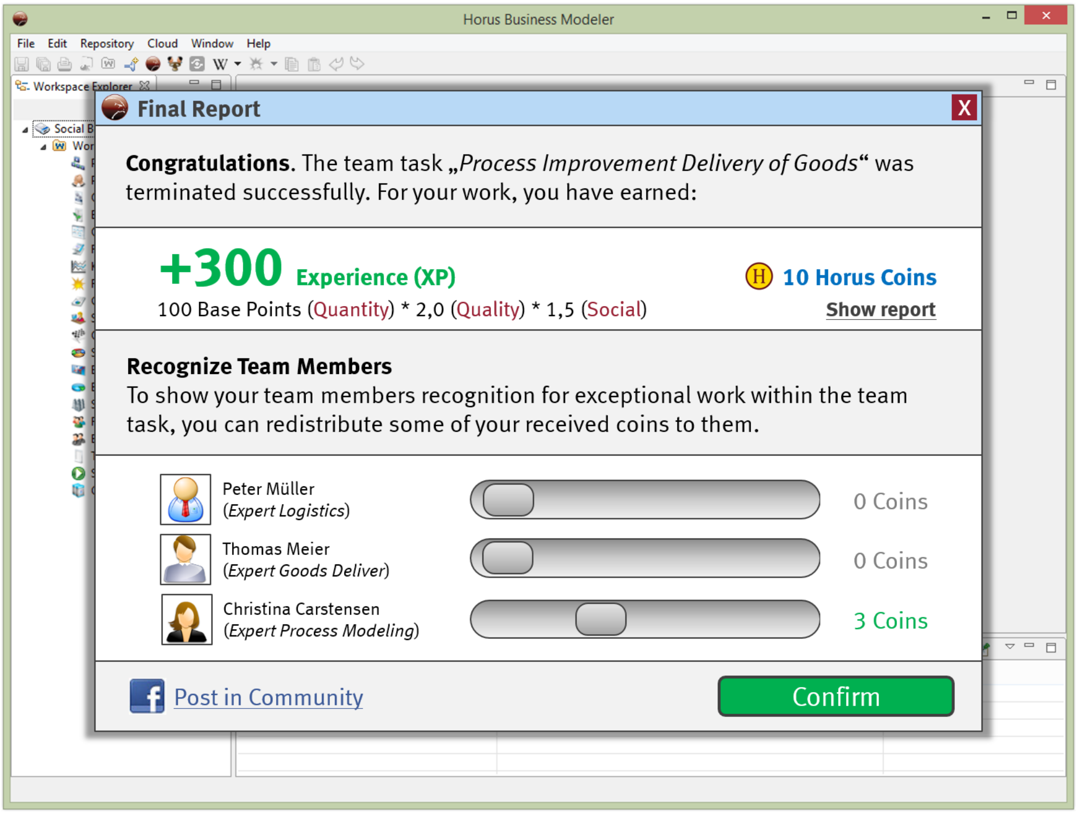


Figure 6.5: Gamification design concept: Team task completion.

1999) features a learning campaign in which players are taught basic gameplay aspects such as moving units, attacking enemies, collecting resources, building structures, and researching technologies. This can be compared to learning new aspects of the syntax of a modeling language via tutorial tasks. However, many modern games do not contain explicit tutorials that can be clearly distinguished from the main game. Instead, a modern approach to tutorials consists of an onboarding phase during which players are successively taught new game mechanics that is embedded within the main story or scenario.

Quiz tasks

During a quiz, users are presented with multiple questions from a server-side catalog that they have to answer correctly within a limited amount of time. Questions can for instance address conceptual aspects of the Horus method, the theory behind business process modeling, the syntax of Petri nets, or the contents of specific models. Generally speaking, all types of questions

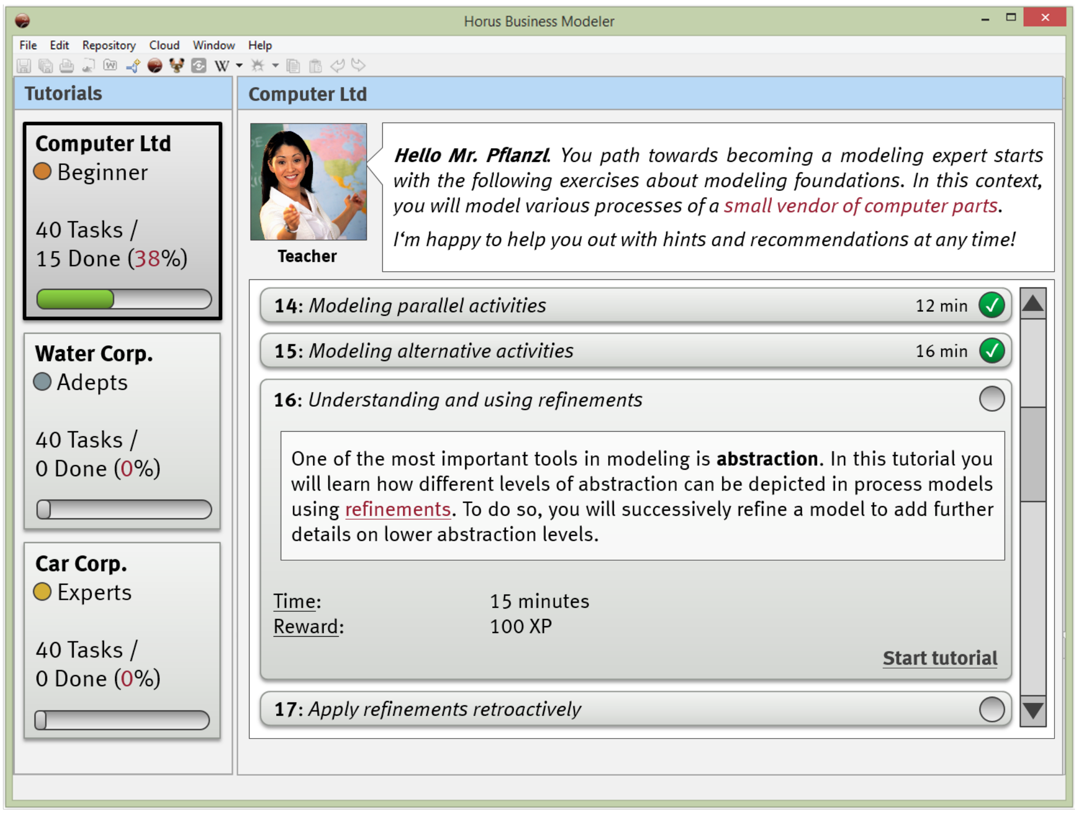


Figure 6.6: Gamification design concept: Tutorial list.

are possible whose responses can be evaluated automatically, such as yes/no questions, multiple-choice questions, puzzle pictures in which users have to search for modeling errors, or answering questions about a business process based on its model. At the end of the quiz task, modelers are granted an amount of experience points influenced by the correctness of their answers and the elapsed time.

Dashboard

The gamification dashboard is the main entry point for users starting a new modeling session in the gamified HBM. Therefore, it presents users with an overview of the tasks that they can perform and recent events in the repository that concern them. As shown in Figure 6.8, examples include notifications about badges that have recently been unlocked, ratings for models created by the user, and team tasks with impending due dates. Furthermore, shortcuts to gamification-specific settings and help contents can be provided. Depending

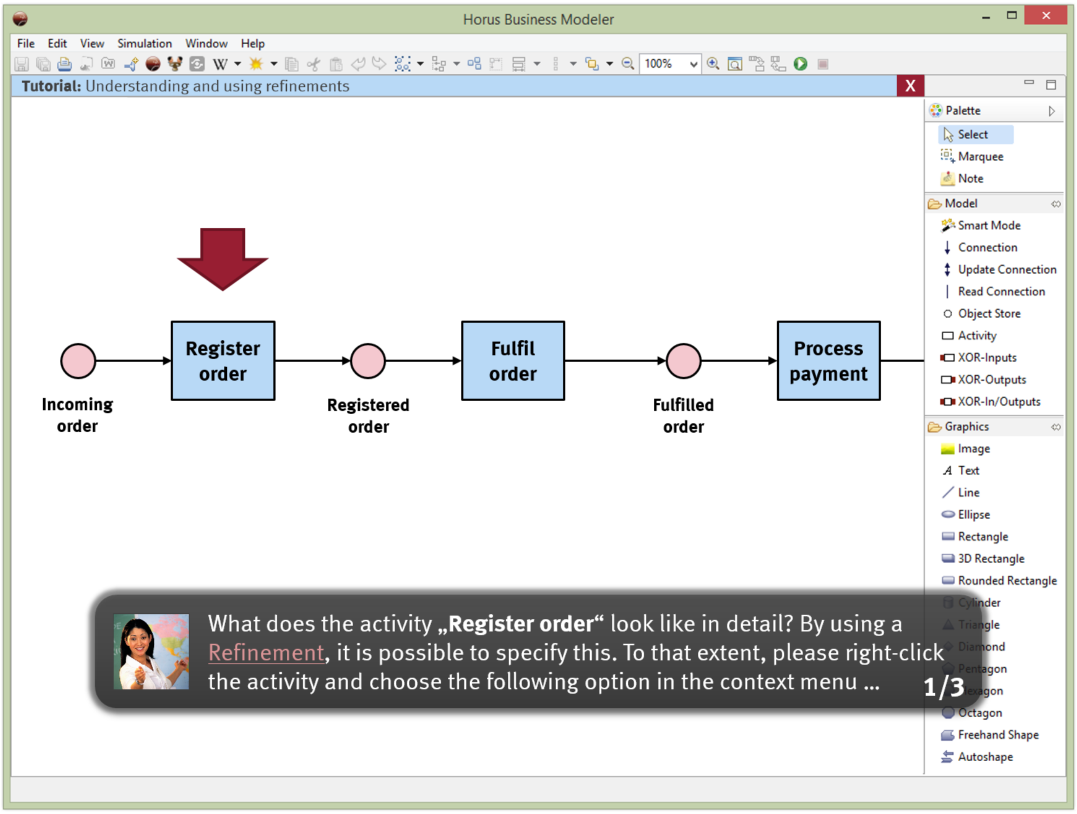


Figure 6.7: Gamification design concept: Individual tutorial.

on the exact contents that are shown, the dashboard can also make a significant contribution to *awareness* within the HBM, a concept from Computer-Supported Cooperative Work (CSCW) research denoting the "understanding of the activities of others" [DB92, p. 107] that enables users to perform meaningful actions in group work and that is of special importance in shared workspaces [GGR96]. In terms of a game, the dashboard can be compared to the menu screen typically shown directly after the game is launched.

Introduction

The introduction page displayed in Figure 6.9 is a special version of the dashboard that is opened when a new users logs in for the first time or when he clicks the "help" button on the regular dashboard. The purpose of the introduction is twofold: firstly, it allows users to understand the context in which gamification is used and their roles in the former. For instance, if it is used in an organizational context, users should understand that they can proactively

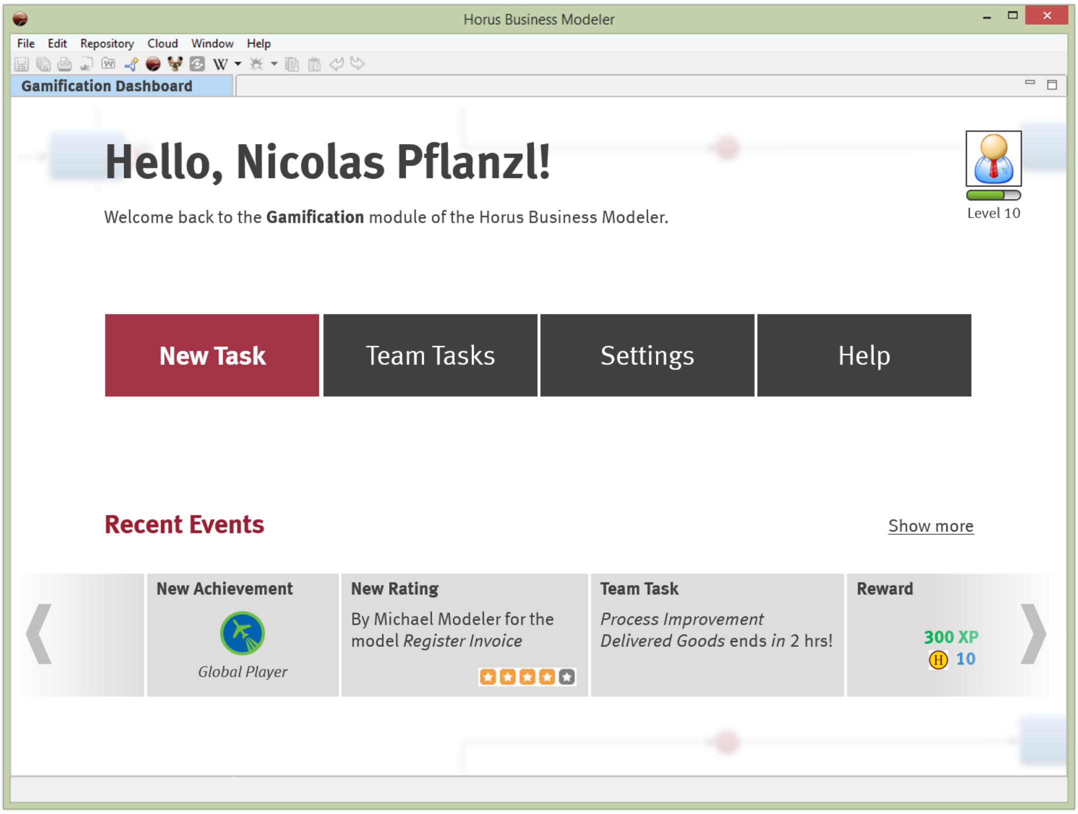


Figure 6.8: Gamification design concept: Gamification dashboard.

influence the concrete manifestation that business processes will have in the future. In this sense, the introduction page can be compared to the manual of a video game. Secondly, the expectations of users regarding business process modeling should be initialized in a positive and motivating way. This can for instance be achieved by granting new users an insight into what they can learn by participating in BPM, how this helps them with the development of new skills, and how they can commence this learning experience.

6.1.2 Rewards

Most of the tasks described in the previous section result in a quantifiable work result that can be used as a basis for providing modelers with feedback and granting them rewards. In the following subsections, six different types of rewards and the conditions for their allocation are discussed.

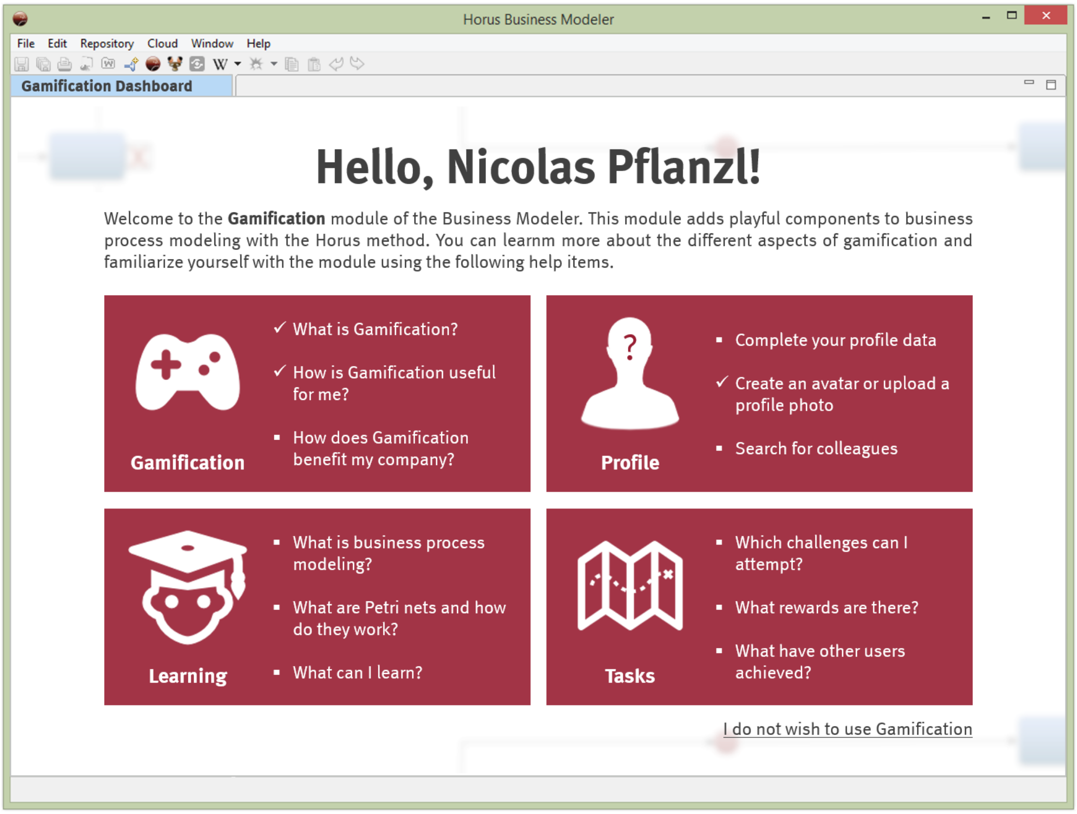


Figure 6.9: Gamification design concept: Gamification introduction.

Points (Score)

Model quality is a property of an individual process model with a standardized value between zero and one and thus does not lend itself as a basis for rewarding modelers. Therefore, points are introduced as an additional construct that users can earn for the work they have carried out and that are amassed over time. To that extent, for each model, a points score is calculated in addition to its quality. For instance, assuming the existence of the 32 quality metrics listed in Table 6.1 and an (arbitrary) number of 10 points that can be earned per metric, all models can have a score between 0 and 320 points. The difference between the number of points that a model has when a user starts working on it and the number of points it has whenever its current state is saved then determines the height of the reward that the modeler obtains. For example, if the model has 250 points at the beginning and the user is able to improve its score to 270 points, he would earn a reward of 20 points that is permanently added to his profile.

If the quality (and thus the score) of a model decreases, this should theoretically lead to a penalty for the user who is responsible for the corresponding change. However, Schell points out that the loss of points is such a painful punishment that it lessens the value of any points that have been earned, and thus such measures are rarely employed in games [Sch08]. Nevertheless, if harmful actions do not lead to any negative consequences, this may prompt users to repeatedly impair and improve a model again to gain more points without actually making valuable contributions. This is a practice called "gaming the system" that should be kept in mind and monitored for once the gamified system has been rolled out [ZC11].

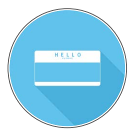
Experience Points and Levels

Whereas "normal" points relate to models, experience points (in short XP) are an indicator for the activity and progress of individual modelers. They can be earned for a variety of different tasks, such as completing quality tasks (see Figure 6.2), participating in team tasks (see Figure 6.5), or successfully completing tutorial tasks or quiz tasks (see Figure 6.7). Therefore, the overall number of experience points a user has earned may be seen as a proof of the modeling skills he possesses. Upon reaching certain experience milestones, users advance levels, e. g., from level 1 to level 2. Thus, levels represent an aggregated measure of experience and further simplify the assessment of a modeler's skill level. Therefore, levels can be used as a criterion during the creation of a team task to specify the minimum skill level that an applicant must possess to participate. Furthermore, experience points can be used as the main criterion for ranking users on the so-called leaderboard (see Section 6.1.4), thus promoting competition and creating incentives to perform additional modeling activities to surpass other modelers. Finally, levels can also serve as a gateway for unlocking privileges, another type of rewards discussed below. Experience and level systems are a staple of many games, in particular RPGs such as Diablo III and Final Fantasy IX. However, they are nowadays also found in many other genres, such as FPSs.

Badges

As outlined in Section 3.4.1, badges (also called achievements) are virtual rewards that players can earn by fulfilling certain conditions that are ancillary to the primary goals of a game. They can serve many different purposes, including goal setting, instruction, reputation, status and affirmation, and group identification [AC11]. Transferred over to the context of the HBM, badges can be earned by performing specific actions within and without the model editor that are deemed to be particularly noteworthy, for instance through behavior that is considered to be beneficial for model quality. The badges that have been earned by a user are prominently displayed on his profile as shown in Figure 6.10. Depending on the difficulty of the conditions that must be met, badges can be subdivided into different classes, such as bronze, silver, and gold badges. Furthermore, some badges successively build on and thus replace each other by having the same conditions, albeit with different quantifiers. Finally, besides normal badges with clearly communicated unlock conditions, there can also be "secret" or "hidden" badges whose existence is only revealed to users after they have met their requirements. Building on the badge description framework proposed in [HE11], Table 6.2 specifies the signifiers of 22 distinct and 9 successor badges, resulting in a total number of 31 sample badges that users can earn. As the completion logic of these badges can be derived from their descriptions, the former is not provided here in detail but can be found in the appendix in Section A.1. Further contained in Section A.2 of the appendix is a table specifying how the positive effects of the proposed badges are expected to manifest themselves following the classification by ANTIN AND CHURCHILL [AC11].

Table 6.2: List of badges in the Horus gamification concept.

ID	Level	Name	Description	Visual
Completion of user profile				
1	①	I know your name	You have provided your first and last names.	
	②	I know where you live!	You have provided all basic user information.	

Continued on the next page ▷

Table 6.2 (contd.)

ID	Level	Name	Description	Visual
2	①	Look at me!	You have uploaded a picture of yourself. So that's what you look like!	
3	①	Reachable	You have provided 5 contact methods.	
4	①	Capable	You have defined 5 personal skills.	
5	①	Endorser	You have endorsed 10 skills of others. How nice of you!	
General user activity				
6	①	New User	Welcome to the Horus Business Modeler!	
	②	Power User	You have logged in for the 100th time.	
7	①	Returner	Welcome back! You have logged in at least three days in a row.	
	②	Metronome	Welcome back! You have logged in at least five days in a row.	

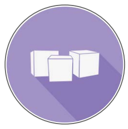
Continued on the next page ▷

Table 6.2 (contd.)

ID	Level	Name	Description	Visual
	③	Junkie	You again! You have logged in at least seven days in a row.	
8	①	It's your birthday	Your account has just turned one.	
	②	It's your birthday... again	Your account has just turned two.	
	③	It's your birthday... yet again	Your account has just turned three.	
9	①	Christmas Modeler 2015	You logged in on Christmas day 2015. But... why?	
10	①	New Year's Modeler 2016	You logged in on New Year's Day 2016. Happy new year!	
Activities in the workspace explorer				
11	①	Process Model Creator	You have created 10 process models.	
	②	Process Model Creator	You have created 100 process models.	

Continued on the next page ▷

Table 6.2 (contd.)

ID	Level	Name	Description	Visual
12	①	Amateur Constructivist	Building things feels so good. You have created at least 10 models of different kinds, but deleted less than 5 models at the same time.	
13	①	Amateur Destructivist	Destroying things feels so good. You have deleted at least 10 models of different kinds, but only created 5 at the same time.	
14	①	Zen Novice	You have created at least one process, object, and organization model, and one role.	
	②	Zen Master	You have created at least one model of every kind.	
Process modeling				
15	①	You Shall Not Cross	You have removed at least 10 edge crossings.	
	②	You Shall Not Cross	You have removed at least 100 edge crossings.	
16	①	Object Type Completionist	You have added 30 object type references to process models.	
17	①	Role Completionist	You have added 30 role references to process models.	

Continued on the next page ▷

Table 6.2 (contd.)

ID	Level	Name	Description	Visual
18	①	Babbler	You have written 10000 characters of textual documentation.	
19	①	Snake Man	You have created a model with a diameter of 30 nodes or more. Can you still understand it?	
20	①	Megalomaniac	You have created a model with 50 elements or more. Don't you think you should split that up?	
21	①	Pixel Pusher	You have moved objects around over 1,000,000 pixels. Such movement!	
22	①	Points Collector	You have earned 10,000 points while modeling.	

Horus Coins

Many games contain currency systems that enable players to amass coins, gems, or similar objects by overcoming certain challenges and to exchange these units for in-game items and other rewards. Thus, in contrast to many score and experience systems, points of this type can not only be accumulated, but also spent again by design. In the context of the HBM, Horus Coins can be earned by participating in team tasks, or theoretically also for any other activity discussed in Section 6.1.1. In a special shop, users may exchange their coins for virtual objects that they can use to customize their avatars and for other virtual or monetary rewards defined by the responsible organization, although the latter is often discouraged due to its potential negative impact on intrinsic motivation under certain conditions [LGN73, CP94, RD00c]. Furthermore, they may also invest them to initiate event tasks or to "buy" into team tasks for which they may not yet fulfill the necessary preconditions.

Virtual Objects
Another category of rewards are virtual objects that can be used to individualize user avatars. Consequently, such objects can therefore be virtual pieces of clothing, accessories, or other items—possibly with corporate branding by Horus or the responsible organization. Whereas a small set of basic objects may already be available to users when they start using the gamification module, further objects can be bought with Horus Coins.

Privileges
Finally, users can also earn privileges that extend the set of possible actions that they are able to perform. This type of reward can be found in a number of gamified products such as Stack Exchange (Stack Exchange, 2009) and YouTube Heroes (YouTube, 2016), in which long-time users who have demonstrated their expertise and have repeatedly shown to be reliable are granted special rights for community management and content moderation. In the context of the HBM, privileges can for instance be earned by accumulating enough experience points to reach a particular level, or by purchasing them using Horus Coins. Sample privileges are the right to define event tasks, the right to create team tasks, the right to buy into a team task without possessing the specified skills, or the right to individualize the user profile with custom background colors or images.

6.1.3 Players

In the context of the gamified HBM, “players” are the human individuals who participate in the “game” of business process modeling as part of a larger modeling project. Such a project may for instance be any of the scenarios discussed in Section 2.2, a Social BPM lab [CCL+13, ACUV16], or a modeling case study conducted as part of a university lecture. As outlined in Chapter 4, the knowledge and skill set of participants pertaining to BPM and the modeled domain can be very heterogeneous, ranging from complete novices to seasoned experts. Furthermore, these actors may either work alone (e. g., in quality tasks or tutorial tasks) or as a group (i. e., in team tasks). The gamification features presented in this section are designed to increase the visibility of any relevant information about these actors and to serve as a record of their achievements.

Profile

Every user has his own profile on which he can provide personal information, including a real name, a profile picture, interests, and various contact methods (e. g., email, telephone, Skype), to increase his visibility, identifiability, and reachability. Additionally, the profile allows for defining references to Horus-specific entities that are related to oneself, such as an organization unit, a role object or an employee object. Likewise, users can also indicate their areas of expertise, pertaining for instance to BPM skills, other technical skills, or areas of domain knowledge. Once again, this may also be done by referencing entities in appropriate skill models. Following the example set by the professional social network LinkedIn, other users can support the claims made by a modeler by *endorsing* his capabilities [BHV+14]. Hence, the number of endorsements a user has received for a certain expertise can also be interpreted as an indicator for the strength of this claim. Lastly, the profile also documents the achievements of each modeler, including the total number of experience points he has earned and his resulting modeler level, the number of badges he has unlocked, and the amount of Horus Coins he possesses. An illustrative mockup containing most of the aforementioned elements can be seen in Figure 6.10.

Avatar

In addition to a profile picture, users can further influence their internal appearance within the HBM by creating a digital avatar with an optional pseudonym. Generally speaking, an avatar is a strongly simplified representation of a modeler that serves as his delegate and can be identified with his actions inside the software. To that extent, avatars can be used as a replacement for profile pictures wherever appropriate, such as in user profiles or the leaderboard. When creating an avatar, users are given the possibility to configure their appearance regarding facial features, figure, clothing, and other visual properties. Special items that allow customizing avatars even further can be bought with Horus Coins and represent the main intended use of the latter. Lastly, the appearance of avatars can also adapt automatically over time to reflect the achievements of a particular user; for instance, the avatar of a level 20 BPM expert may look distinctly different from that of a level 1 novice. It should be noted that providing a sophisticated, yet easy-to-use configuration tool for avatars is a challenging endeavor in its own right, and thus should not be underestimated. Examples

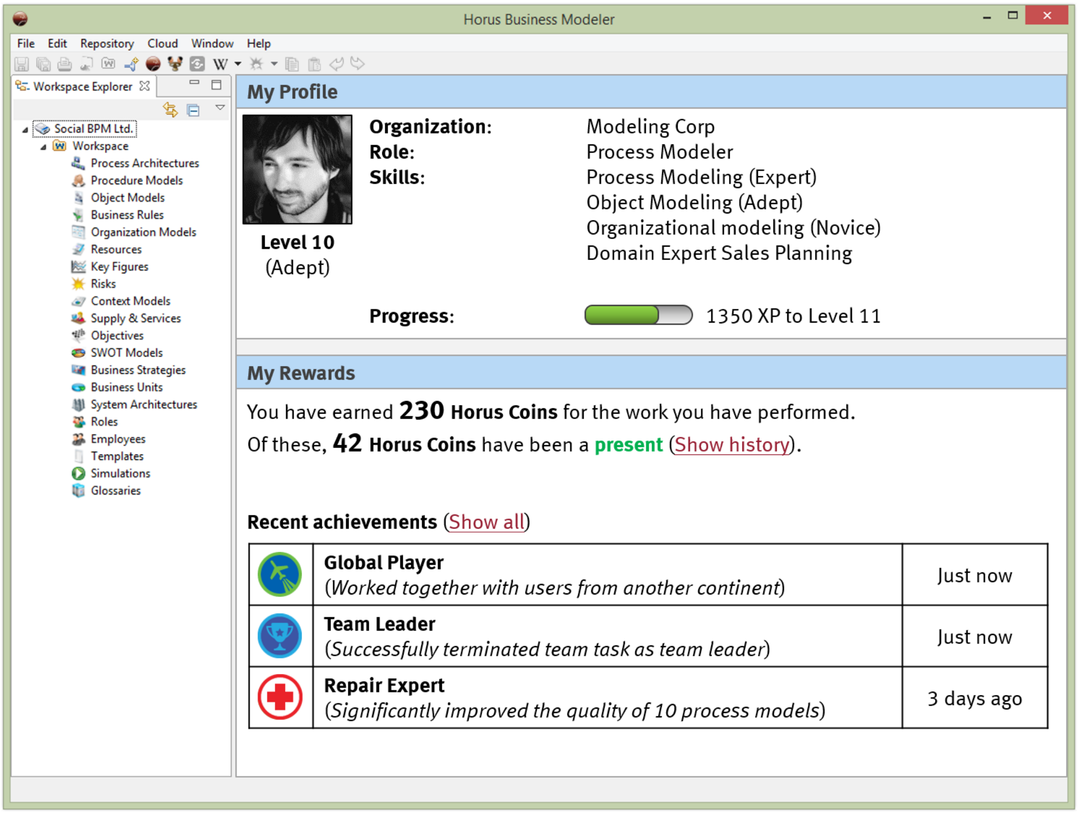


Figure 6.10: Gamification design concept: User profile.

for the use of avatars in games are numerous, especially in the context of RPGs such as The Elder Scrolls V: Skyrim (Bethesda, 2011), where customization allows players to model an idealized form of themselves through which they can interact with the game's world [Sch08]. To a lesser extent, avatars are also used in gamification, especially as a means for self-expression [TLB14].

Skill Tree

Skill trees are a common element of many RPGs such as The Elder Scrolls V: Skyrim and The Witcher 3: Wild Hunt (CD Projekt RED, 2015), where they visualize the sets of abilities that player characters can earn over the course of the game, including physical attacks, magical spells, and non-combat crafting skills. They can also be found in some gamified applications; for instance, the language lessons in Duolingo are presented in a similar manner. While not always adhering to a strict tree-structure, most skill systems are designed in a hierarchical manner so that some skills must be learned as preconditions for others. Carried over to the context of business process modeling, the set of abilities is comprised of, e. g., the elements of the utilized modeling language and the functionality of the employed software. Consequently, users can claim to have learned a skill once they have (repeatedly) demonstrated their capability to use it productively. By arranging these elements hierarchically, multiple skill trees for conceptually similar abilities can be fashioned. A dependency in a tree instance can for example be that modelers are required to learn modeling a sequential flow of activities before being introduced to parallel and alternative flows. To facilitate this learning journey, clicking a particular tree node automatically directs users to relevant tutorial and/or quiz tasks. The use of distinctive icons or colors can help users to distinguish between skills they have already learned, skills that they are ready to learn, and skills not available yet due to unsatisfied preconditions. Overall, the main purpose of the skill tree lies in making the experience of learning process modeling more transparent by presenting users with a visualization of their progress. A sample illustration of the skill tree for the HBM is shown in Figure 6.11.

6.1.4 Competition

Many gamification designs put a strong emphasis on enabling competition and conflict between its target users. By allowing users to compete with each other, these applications aim to appeal to the need of some to outperform others, which then assumes the function of an extrinsic motivator. This type of incentive may not be equally effective for all users, as research has for instance shown that compared to men, women are more averse to competition [CG09]. For this reason, competition is often discussed in the context of specific player typologies, most notably the "killer" type in the Bartle's widely-used MUD player types [Bar96]. The following paragraphs discuss opportunities for enabling competition and conflict within the gamified HBM.

Leaderboard
On the leaderboard, the performance of modelers using the HBM on a particular repository is ranked and visually depicted. In the most straightforward case shown in Figure 6.12, the leaderboard is based on the aggregated number of experience points that users have earned. However, creating alternative rankings for Horus Coins held or spent, or the number of badges earned is

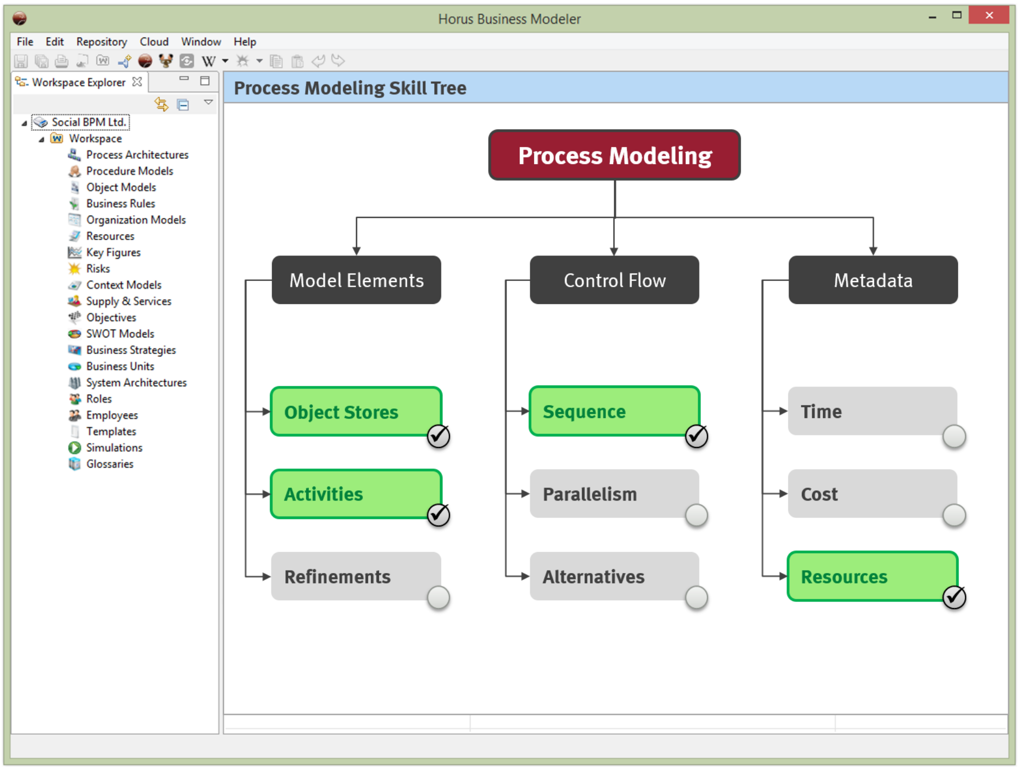


Figure 6.11: Gamification design concept: Skill tree.

possible as well. Furthermore, the leaderboard can be filtered according to temporal criteria (e. g., to limit it to points earned in the last week) and network criteria (e. g., to show a ranking of all users in the immediate network of a focal user). The leaderboard can be expected to have a positive impact on the motivation of users who wish to outperform others and who perceive themselves as having a realistic chance to do so. Otherwise, it may even have negative effects, thus making it important to let users decide themselves whether they wish to participate in the ranking. Sample measures that may diminish such undesirable consequences include anonymizing the leaderboard and limiting the area visible to users to their immediate neighbors.

Profile Comparison

As a complement to the leaderboard, users are given the possibility to compete with others more directly by using the profile comparison feature. This includes a simplistic, quantitative summary, but further expands upon it by making use of the entire breadth of the data that using the gamification module creates.

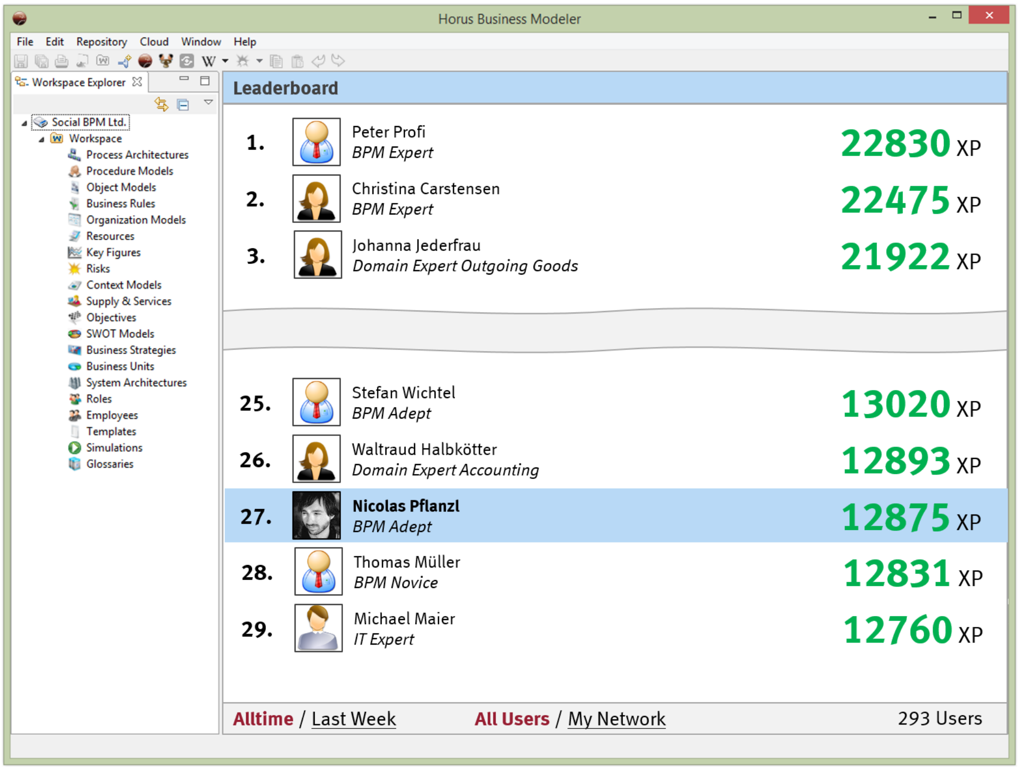


Figure 6.12: Gamification design concept: Leaderboard.

Consequently, the basis of the profile comparison is formed by, e. g., the overall number of experience points, levels, and Horus coins, the list of unlocked badges, the number of obtained experience points grouped by quality task, the ratings received by other users, and other quantifiable aspects of modeling activity. While not very common in the context of games, a profile comparison feature can for instance be found on the Xbox Live service, where users may compare their achievements with those of others on a game-by-game basis.

6.1.5 Interaction

Due to the highly interactive and collaborative nature of Social BPM, functions that allow users to connect and communicate with others are of considerable importance. Furthermore, they are also required by team tasks, and are thus an essential component of this gamification design concept. Additionally, some of the other features described so far can also be seen as forms of interaction, for instance leaderboards and profile comparison. In the following, some additional means for communication are described.

Relationships
As in social networking Web sites, users are given the possibility to maintain their relationships to friends, colleagues, subordinates, and superiors. This can either be done explicitly by sending and receiving contact requests to resp. from others, but also implicitly in the context of team tasks. Unlike in personal social networks, it should be ensured that the utilized terminology properly reflects the professional nature of relationships in Social BPM, for instance by using the term “contacts” rather than “friends”. The idea of specifying relationships with others is very common in modern digital games with online components, and is typically connected to advanced features for awareness and cooperation. For instance, upon adding another player as a friend in Diablo III, it becomes possible to quickly start a chat with the other party, invite them to play together, or examine their current status in the game.

Communication
As part of the gamification module (and possibly also beyond), users are given the possibility to communicate with one another synchronously or asynchronously in textual form, for instance via chats and comment areas. This can be done, e. g., in a private conversation between two users, while collaboratively modifying a process model, or while working together on a team task as shown in Figure 6.13. Any non-private conversation is always connected to a particular Horus entity, such as a repository, a workspace, a team task, a model, a region of a model, or an individual model element. Once a conversation has fulfilled its purpose, it can be closed and will then be discarded. Besides these explicit forms of communication, social tasks such as rating and commenting can also be seen as an implicit type of conversation.

6.1.6 Resources

In the context of a game, resources are assets that are only available in limited quantities, and that players require for overcoming the challenges of the game [Ful08]. For instance, in many RPGs such as Castlevania: Symphony of the Night, Diablo III, and Final Fantasy IX, the concept of *health points* that can be

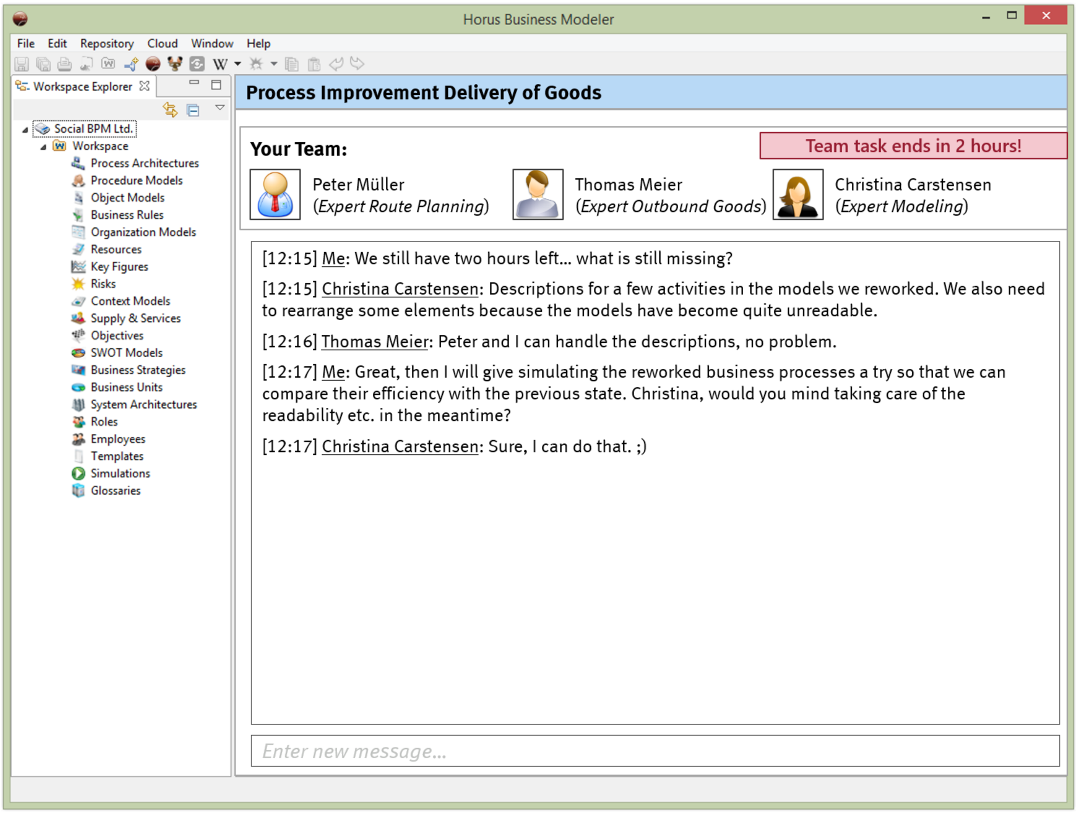


Figure 6.13: Gamification design concept: Team task chat.

diminished, e. g., by enemy attacks or traps, exists. If their number falls to zero, players are typically confronted with a “game over” screen and are required to continue from the last save state. Other games such as Super Mario World limit the time players have at their disposal to finish individual levels. Limiting resources may serve many different purposes, including the regulation of the difficulty of a game, forcing gamers to advance through a level carefully and prudently, or inversely, putting players under time pressure to increase the error-proneness of their actions [Ful08]. In the context of business process modeling, resource limitations may be introduced as a game element as follows.

Model Elements

As a consequence of the skill tree described in Section 6.1.3, the set of model elements from which users can draw while creating process models can be treated as a resource with limited availability. While this may not have as wide-reaching consequences for modeling languages with relatively few types of model elements—such as Petri nets as compared to BPMN with over 100 different constructs—it may still prevent users from being overwhelmed with too much choice. Thus, such measures have already been proposed in previous research not connected to gamification, e. g., [Rit10b] and [Bra13]. Depending on the degree of limitation, three different scenarios can be distinguished: 1. *Full limitation*: Model elements only become available once they have been unlocked through a corresponding tutorial or quiz task. Some activities lying outside the concrete syntax of the modeling language may still be allowed, such as writing comments or drawing graphical objects. 2. *Partial limitation*: A core set of modeling elements is always available; advanced elements first need to be "learned" before they become available. This is a compromise resulting from the observation that many modeling languages offer a large variety of different notational elements of which only a small subset is regularly used in practice (cf. [zMR08] for a discussion on BPMN). 3. *No limitation*: All model elements are always available and the skill tree serves no additional purpose. Limiting the set of possible actions that users can carry out in one of the first two manners is common in many games where this lets the full complexity of the gameplay and the game mechanics unfold over time, thereby slowly easing new players in. Finally, it should be noted that any limitations should always be optional, as enforcing such restrictions upon modeling experts serves no purpose other than to prevent them from working effectively.

Time

Time as a limited resource has already been discussed in previous parts of the gamification concept, and thus only a brief summary will be presented here. Firstly, the initiator of a team task can specify a due date to put participants under pressure and motivate them to prioritize it. Secondly, as the name already indicates, the distinctive feature of event tasks also is that they exhibit a limited temporal validity. Lastly, tutorial tasks and quiz tasks also have a predefined time limit that must be respected by users for successful completion.

6.2 Concept Lenses

The purpose of this section is to outline how the various parts of the proposed gamification design are expected to contribute the goals defined in Section 4.3.1, namely improving model quality, increasing motivation to model, and improving training and education. To that extent, the following subsections view the concept through three different *lenses*, i. e., a quality lens in Section 6.2.1, a motivation lens in Section 6.2.2, and a training lens in Section 6.2.3. The concept of lenses is adapted from game design literature, where the term refers to different perspectives from which a game design can be viewed [Sch08]. It has previously been adapted to the context of gamification by Deterding, who proposed a new lens (in addition to those defined by Schell [Sch08]) specifically for the design of gamified applications [Det15]. Lenses have also been used in information visualization as a tool for viewing a complex design (i. e., a visual representation) through a viewport that emphasizes particular details while abstracting from others [BSP+94, vLKS+11]. Synthesizing these approaches, lenses will in the following be used to describe the gamification design from various perspectives while omitting anything deemed irrelevant or secondary for a particular lens.

6.2.1 Quality Lens

The subset of features included in the gamification concept that are expected to have a positive impact on model quality are depicted in Figure 6.14. In particular, it is argued that the proposed design gives users the following facilities for creating and maintaining high-quality models:

- As their name already indicates, quality tasks are specifically geared to quality maintenance and improvement. The real-time feedback that users receive enables them to not only create new models with high quality from scratch, but also to improve the quality of already-existing models. Quality tasks may address any aspect of a process model and are only limited by the feasibility of implementing a measurement procedure for the respective quality metric. However, it should be noted that not all users may be able to operationalize all metrics with the same effectiveness. For instance, a concept such as "edge crossings" is easier to comprehend than "control-flow complexity", and thus more significant effects can expected for the former.

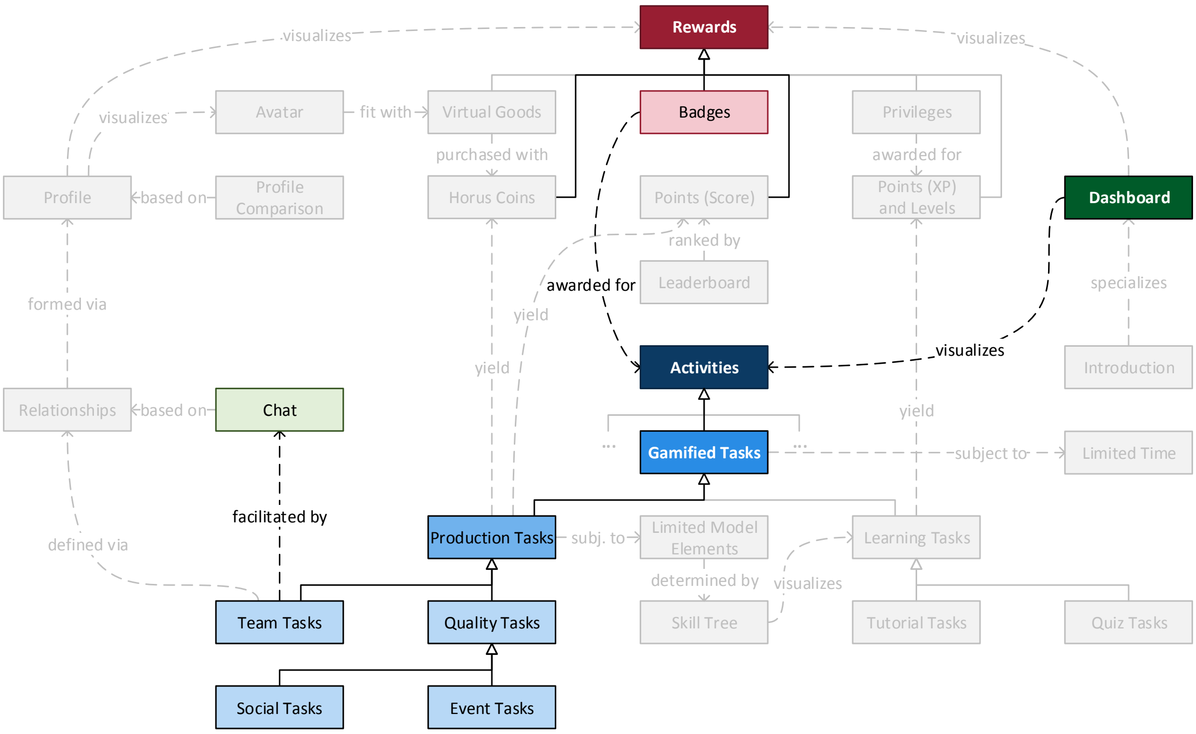


Figure 6.14: Gamification design lens: Model quality.

- Compared to quality tasks, team tasks are much more flexible as their goals are specified by the humans who carry them out instead of predefined rules. Similarly, the success of a team task is evaluated by its participants rather than algorithmically. Overall, this means that team tasks can contribute to aspects of model quality that are difficult (or even impossible) to measure automatically, such as, in terms of the SEQUAL quality framework (see Section 2.6.2), semantic, social, and deontic quality. However, this also means that feedback related to these aspects must be provided by the participants of the team task through the use of the chat function and will therefore not be available in real-time.

- If the reward conditions that must be met to unlock a particular badge are connected to quality metrics, they may also contribute to model quality. This is the case, for instance, for the badge “You Shall Not Cross”, the two “Completionist” badges, and the badge “Babbler” presented in the previous section. Assuming that modelers are able to internalize the behavior that has allowed them to earn a badge, the positive effects may even persist when no further achievements are available.

- Except the increased reward size, event tasks are equivalent to quality tasks and can thus be expected to impact model quality in the same manner. However, due to the increased visibility that event tasks receive from being presented directly on the dashboards of all users, they may be addressed with higher priority and by a larger number of users.

- Social tasks such as ratings and comments do not have a direct impact on model quality, but may prompt further modifications that do.

6.2.2 Motivation Lens

The gamification concept does not address motivation as a singular construct, but from different perspectives that will be addressed separately in the following. These perspectives are extrinsic motivation, self-determination and intrinsic motivation, and education and training.

Extrinsic Motivation Lens

The subset of features included in the gamification concept that are expected to have a positive impact on extrinsic motivation are depicted in Figure 6.15. Most significantly, all of the six types of rewards that users can earn while working with the HBM—points from quality tasks, experience points and levels, badges, Horus coins, virtual goods, and privileges—provide modelers with external goals to strive for. Furthermore, users who are interested in competing with others may draw additional motivation out of the leaderboard and the profile comparison feature, which are designed for this particular purpose. It should be kept in mind that this may come at the cost of intrinsic motivation, in particular for expected rewards handed out for the completion of high-interest tasks [DKR99]. However, current gamification research has yet to fully understand the impacts of rewards such as those included in this design concept on intrinsic and extrinsic motivation, and thus the direction and extent of their effects cannot be fully predicted in advance (see Section 3.5.1).

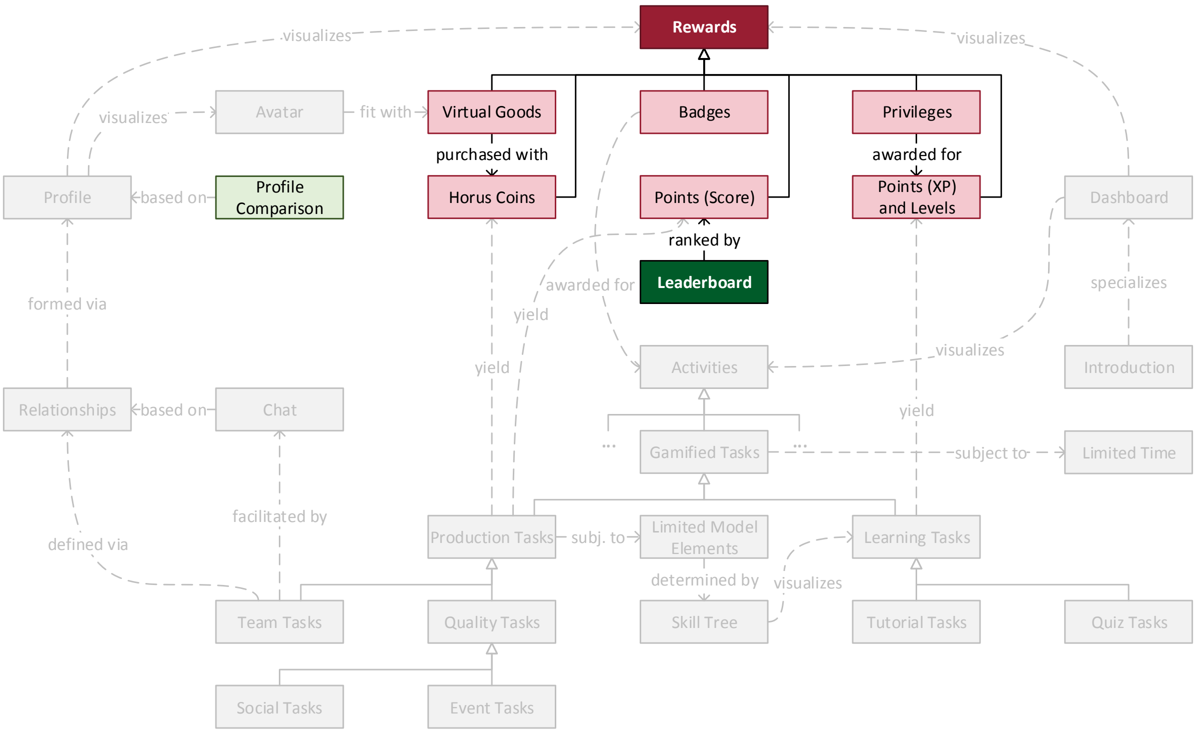


Figure 6.15: Gamification design lens: Extrinsic motivation.

Self-determination Lens

The subset of features included in the gamification concept that are expected to have a positive impact on self-determination, and thus possibly intrinsic motivation are depicted in Figure 6.16. In particular, it is argued that the following features and sets of features contribute to the fulfillment of needs for autonomy, competence, and relatedness as follows:

- Generally speaking, almost all of the rewards that users can earn are expected to provide feelings of competence by providing tangible indicators for the quantity and quality of conducted work. It should be noted that this may come at the cost of autonomy if those rewards are perceived as controlling rather than as informative, positive feedback. A special role is played by badges, which can work in many different ways and may thus also contribute to autonomy and relatedness [AC11]. For instance, whereas an unexpected, hidden badge may strengthen the conviction that a user is performing valuable work in an autonomous fashion, sharing a badge with others may strengthen perceptions of being part of a group with shared values and beliefs.

- Virtual goods that can be purchased with Horus coins together with the possibility to create and customize one's own avatar provide users with elements of choice, and may thus contribute to perceived autonomy.

- As the profile page provides a record of everything a user has achieved in the HBM so far (e. g., quality points received, badges unlocked, levels earned), it can be expected to allows modelers to feel competent about their skills and abilities. Furthermore, viewing another user's profile may satisfy the need for relatedness if the two individuals are acquainted or have received similar rewards. Additionally, using the profile comparison function may enhance perceived competence if a user has better performance figures than the other party. Lastly, the use of avatars may increase relatedness as they allow making connections between user names and visual appearances.

- All of the gamified tasks—but most importantly quality and learning tasks—provide users with concrete challenges, whose successful completion can be expected to enhance feelings of perceived competence. This holds especially for those tasks whose difficulty scales with increasing skills of modelers, such as tutorials and quizzes. Quality tasks may further contribute to feelings of self-determined action, as users have considerable choice (i. e., 32 quality metrics as described in the concept above) regarding the quality aspects of a model they would like to focus on. The same cannot be said for event tasks, where the only choice is to carry them out or not. Social tasks may also satisfy the needs for autonomy as users must decide by themselves when, how, and what they comment, rate, or tag, and relatedness due to the fact that they involve other users and their work results.

- The underlying idea of team tasks is that the users decide which modeling activities should be carried out, and which goal should be achieved, and which individuals are required for that purpose. Furthermore, at the end of the team task, the participants themselves rather than

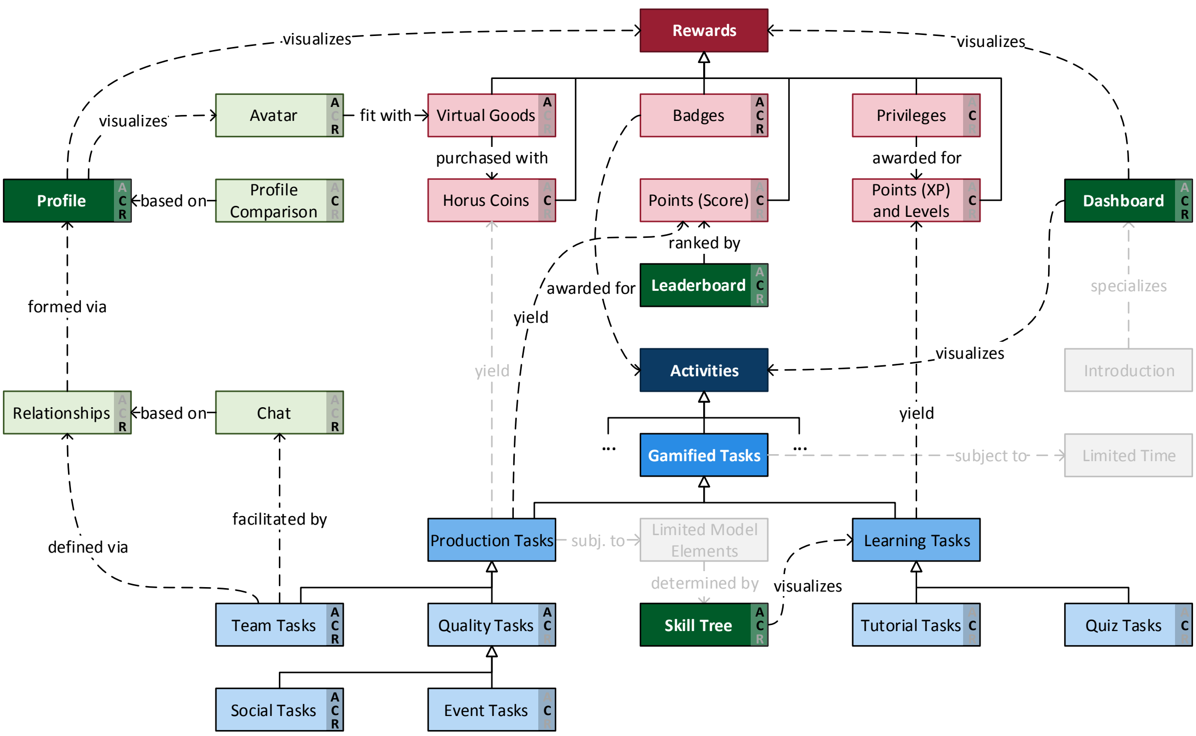


Figure 6.16: Gamification design lens: Self-determination. Letters inside boxes indicate expected contribution to autonomy (A), competence (C), and relatedness (R).

an algorithm evaluate whether it has been successful. Due to this design, team tasks can be expected to have a considerable impact on the satisfaction of relatedness, especially together with the possibility to maintain relationships and communicate with others.

- Similarly to the profile, the skill tree serves as a record of achievements for learning tasks and can thus be expected to provide feelings of competence, especially with an increasing number of successfully completed tutorials and quizzes. Furthermore, the skill tree is intended to be nonlinear by design, so that users can define their own pace and order for learning about process modeling and the functionality of the HBM. Thus, it is argued that the skill tree further satisfies the need for autonomy.
- The leaderboard is intended to enable competition between users, and thus it may allow users to feel competent if they rank highly, or at least higher than other users that are of particular interest to them. Inversely, feelings of competence may be diminished for the global leaderboard (i. e., without any limitations regarding time or users) if users perceive themselves as incapable of improving their rank with realistic effort.
- The dashboard contains a list of recent events affecting a user (and possibly his network), which may, in the case of rewards, contribute to perceptions of one's own competence. Furthermore, if these events are related to team tasks, friends and colleagues, or ratings and comments received by others, the need for relatedness may be affected as well.

Flow Lens

The subset of features included in the gamification concept that are expected to impact flow are depicted in Figure 6.17. In particular, it is argued that the proposed design facilitates the occurrence of flow experiences by satisfying their preconditions as follows [CAN05]:

- **Clear goals** are provided by most of the gamified tasks defined in the concept, i. e., achieving a perfect score of 1.0 for quality tasks and event tasks, following modeling instructions for tutorial tasks, and answering questions correctly for quiz tasks. The dashboard further contributes to goal clarity by providing an overview of recent events and open tasks

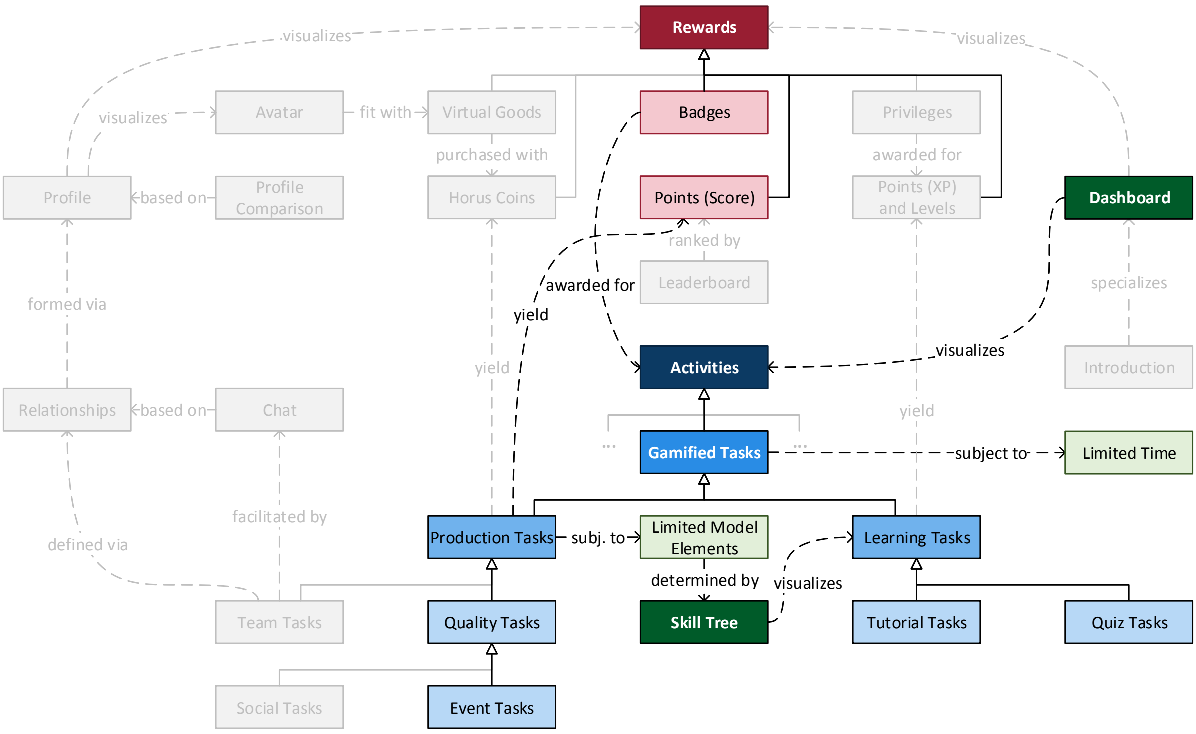


Figure 6.17: Gamification design lens: Flow experiences.

in a repository. Lastly, clear goals can also be provided by badges that are designed for that specific purpose [AC11].

- **Skill-challenge balance**. Various features of the gamification concept ensure that modelers can vary the level of challenge in dependence of their perceived skill. Firstly, tutorials and quizzes are intended to teach fundamental process modeling concepts from scratch, and are thus appropriate for novices. However, the further a user progresses in the skill tree, the harder the respective tasks get to account for the increasing skill of the former. Similarly, if a novice user decides to limit the set of model elements available to him based on his skill tree, this can serve to reduce the complexity of the HBM in accordance with his state of learning. Conversely, a source of challenge is the time limitation that may be imposed on tutorials and quizzes.

- **Immediate, clear feedback** is given to modelers during quality tasks in conjunction with the points score of a model. This feedback is given

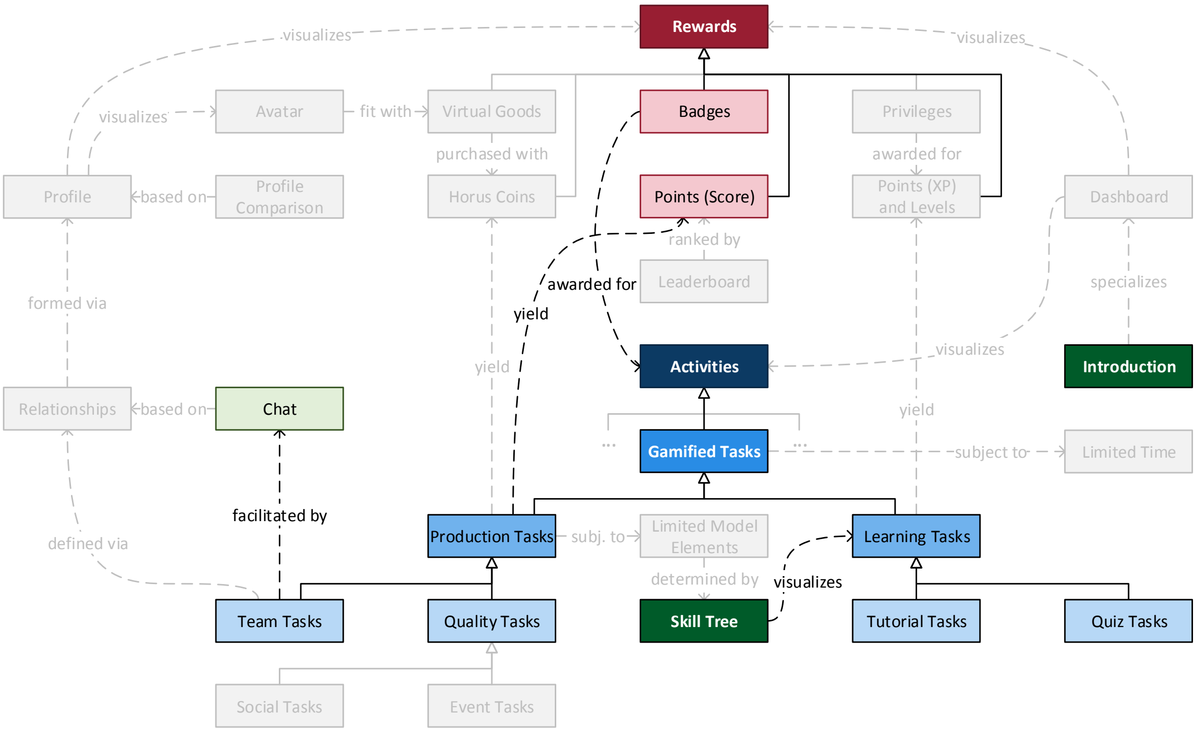


Figure 6.18: Gamification design lens: Training and education.

in real-time, so that any change to the model is immediately reflected in its valuation. Further feedback is provided in the context of tutorial tasks and quiz tasks whenever the modeler has properly carried out an instruction or correctly answered a question. In all three cases, it is always made clear to the modeler what the current status of the task is and whether a particular action has resulted in an improvement or not.

6.2.3 Training Lens

The subset of features included in the gamification concept that are expected to impact training and education are depicted in Figure 6.18. In particular, it is argued that the proposed design facilitates the acquisition of process modeling skills in the following manners:

- The skill tree provides a hierarchical visualization of the concepts that must be learned until an individual can claim to be proficient in modeling processes as Petri nets and using the HBM. Furthermore, it gives quick access to tutorial tasks and quiz tasks which are created for this particular purpose, and should therefore naturally contribute to learning. Thus, the combination of these elements clearly illustrates to modelers what they have already learned, what they still need to learn, and how this can be done.

- Quality tasks educate users about properties of process models that are deemed desirable and the actions that are required to maintain these properties. This only works in conjunction with real-time points feedback, which allows users to directly see the impacts of any modifications they make on model quality.
- As outlined in Section 4.2.3, collaborative modeling sessions involving heterogeneous sets of modelers can facilitate learning through the exchange of domain knowledge and method expertise [RMH13]. Thus, team tasks in conjunction with communication facilities are expected to further contribute to skill acquisition for both modeling novices and experts.
- One of the purposes for which badges can be used is instruction, i. e., teaching users about desirable and highly-valued behavior [AC11]. For instance, the badge "Object Type Completionist" listed in Table 6.2 informs modelers that references between process models and object models can be created and that this is desirable. Inversely, the badge "Megalomaniac" instructs users to split up models with more than 50 elements.
- Basic information about process modeling, the HBM, and gamification is provided to new users on the introduction-variant of the dashboard. This is intended to facilitate the starting phase of using the HBM and ease users into the functionality of the tool.

7 Implementation

The gamification concept presented in the previous chapter can be implemented in a number of ways. Thus, before examining its impacts in Chapter 8, it is necessary to first describe its exact realization so that the likelihood of properly attributing positive and negative effects to the concept or its implementation is increased. To that extent, Section 7.1 first gives a high-level overview of the technological base of Horus, including its client and server side. Next, Section 7.2 describes functionality that was integrated into the HBM as part of this research project, but that is not directly related to gamification. Instead, this code provides services that can also be used for other purposes, but that are nevertheless essential enablers of gamified process modeling. Then, Section 7.3 gives a brief overview of the architecture of Horus, focusing on the extensions that were made on the side of the client as well as the server. Lastly, Section 7.4 presents the actual implementation of the gamification module. In the following, it will be assumed that the reader is familiar with the basic principles of object-oriented programming, common software design patterns (see [GHJV95]), and the Java programming language.

7.1 Horus Foundations

Eclipse[1] is a community of tools, projects, and collaborators that provides, among other things, Integrated Development Environments (IDEs) for many common programming languages, such as Java, C, C++, JavaScript, and PHP. In early releases, Eclipse IDE was developed as a modular IDE that could be extended—but not re-purposed—by third parties [39]. In 2004, the entire application-independent core of Eclipse was extracted and released as the so-called Eclipse Rich Client Platform (RCP), thereby enabling other developers

[1] See: https://eclipse.org. Last accessed: 2017-04-04.

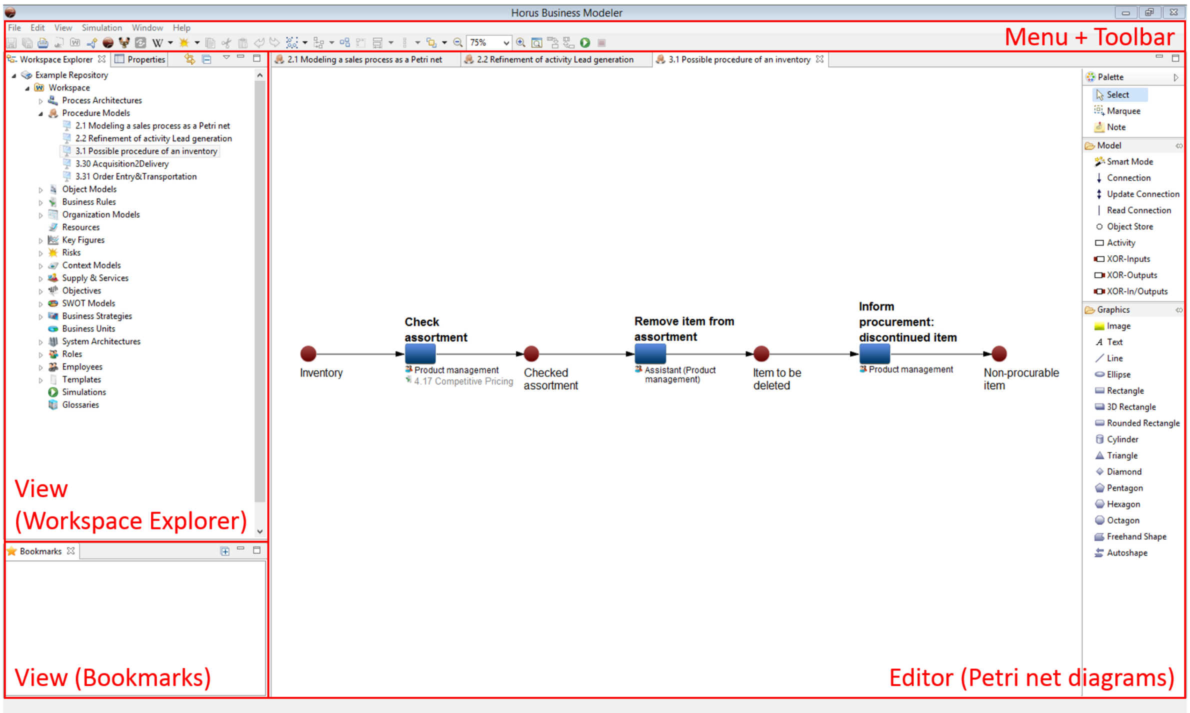


Figure 7.1: User interface components of the Horus Business Modeler.

to reuse the platform and components on which Eclipse is built to create entirely new applications for arbitrary purposes. According to [39], some of the main advantages of using the Eclipse RCP are its stability, the supply of fast and reliable user interface components for various operating systems, its modularity and extensibility, the likelihood of its continued development due to the adoption by large companies such as IBM, SAP, and Google, and its supporting community.

The Horus Business Modeler and its server are developed as two Eclipse RCP applications, their main purpose being the facilitation of business process modeling projects following the Horus method rather than writing source code. Some of the most important parts of the user interface of an RCP application are the *menu*, *toolbars*, *editors*, and *views* [SDF+04]. In the context of the HBM, these elements are depicted in Figure 7.1 and manifest as follows:

- **Menu and toolbar**. Through these components that can be found at the top of the interface, users can quickly access the most important functionality of the HBM, such as adding a new repository connection, creating a new model, or saving changes to the current model. Whereas some functions are available at all times and in all parts of the application, others are context-dependent. For instance, a toolbar icon for the automatic arrangement of model elements is only available in the Petri net editor, and hidden otherwise.

- **Editors** sit at the center of any typical Eclipse RCP application and allow users to create contents within the particular application context. For instance, in the Eclipse IDE, editors provide facilities for writing source code in various programming languages while receiving support through syntax highlighting, code completion, and other convenience features. Similarly, editors in the HBM provide the functionality that is required for creating process models, object models, and the remaining model types enumerated in Section 5.3. For each type of model, the HBM provides a special editor with a fitting palette of model elements and complementary support functions.

- **Views** are any remaining top-level containers of interface elements that are not editors. There are no clear guidelines regarding the possible contents and functionalities of views, and thus this is highly context-specific. However, it can safely be assumed that most Eclipse RCP tools contain a view that is used to search and browse the contents that are managed within the application, and to open them in their respective editors. In the HBM, this is the workspace explorer depicted on the left-hand side of Figure 7.1. Hidden behind the workspace explorer, the properties view allows setting graphical properties and business properties of the currently-selected model element in the active editor. The bookmarks view in the bottom-left corner of the interface provides quick access to the “favorite” models of the current user. The HBM also provides many additional views that are only shown in particular application contexts.

One of the core principles of the Eclipse RCP is that “everything is an extension” [GB04]. Consequently, instantiating the platform only creates a rudimentary workbench that programmers then need to extend by implementing new *plugins*—possibly while using the Eclipse IDE. Such a plugin can add new menu and toolbar items, editors, views, and many other types of user interface elements by adding to so-called *extension points.* An extension point can broadly be described as an interface to which plugins can connect by providing XML code that specifies the desired addition [38]. For instance, the RCP includes an extension point for the creation of new views which requires implementers to specify, e. g., a unique identifier, a name, an icon, and a Java class providing its implementation. Using this data, the Eclipse RCP ensures that the view is initialized and displayed when and where it is required.

Besides just using the extension points that the Eclipse RCP already provides, developers also have the possibility to create new extension points to which they themselves as well as others then can add. The HBM makes extensive use of this mechanism and defines over 20 new extension points for various purposes. For instance, one extension point allows adding new types of repositories to the already-existing local, database, and server connections (cf. Section 5.3). Further extension points allow developers to define new types of models as well as to specify editors which allow creating and modifying instances of the former. The set of available extension points is further expanded by the implementation of the gamification module, particularly enabling the definition of performance measures (see Section 7.2.2), quality metrics (see Section 7.2.3), and badges (see Section 7.4.4). As a result, any developer working on the HBM can, for instance, add new achievements to the gamification implementation without modifying the respective source code by following a simple syntax.

While the characteristics described so far are essential parts of the Eclipse RCP, they only represent a tiny fraction of the entire functionality that the platform provides. However, no additional insights into the RCP and the architecture of Horus are required for understanding the implementation of the gamification module. Therefore, the reader is referred to relevant books such as [SDF+04] and [GB04], or online tutorials such as [38] and [39] for more in-depth information.

7.2 Gamification Foundations

Implementing gamification into any type of application requires the existence of certain supporting features that are not gamification-specific, but serve as its foundation. As these functions were not previously required, and thus unavailable in the HBM, a considerable amount of implementation effort was expended to fit the tool with the required capabilities. This chapter provides a description of these features, specifically event tracking in Section 7.2.1, performance measures in Section 7.2.2, and a model quality framework in Section 7.2.3.

7.2.1 Event Tracking

An important part of any gamified application that in some way incorporates rewards lies in tracking the behavior of its users and making it measurable so that the gamification component can determine which prizes should be awarded to which users, in which quantities, and at which points in time. In the context of Horus, such behavior could for instance be logging into a particular server repository, opening an already-existing business process model, creating a new transition, and then saving and closing the model again. As there was previously no use case for this kind of data within the non-gamified versions of the Horus Business Modeler, user activity was generally not tracked. Instead, when working on a server repository, clients mostly transmit work results (e. g., model objects during a save operation) back to the server, but not the history of actions that have resulted in the former. Thus, to incorporate gamification into the HBM, event tracking capabilities first had to be implemented. In doing so, the following requirements were defined for the event-tracking engine:

- **Extensible**. The set of events tracked by the HBM should not be closed and static, but rather easy to extend with as little effort as possible. This is to account for future developments of the software through which functions may be added, removed, or modified, thereby changing the set of actions that users can perform.

- **Flexible**. The implementation should be generic enough to be usable for a wide variety of different use cases and not specific to gamification. Consequently, it should be easy to register handlers for these events both on the client- as well as on the server-side and handling should be possible in a synchronous as well as asynchronous fashion.

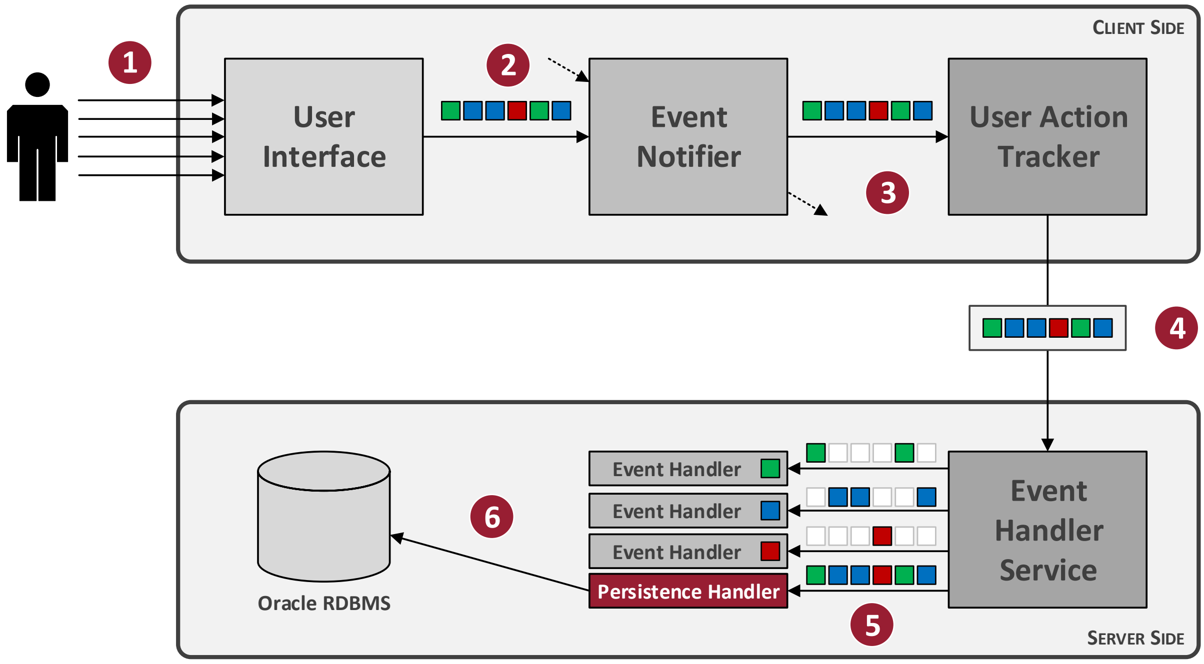


Figure 7.2: Implementation: Conceptual illustration of event tracking.

- **Persistent**. The history of actions performed by users shall be persisted to enable its (re-)examination at later points in time. This enables, for example, *a posteriori* analyses of user behavior, longitudinal analyses of user activity over time, and historic analyses of the state of the system in the past. Focusing on gamification as the use case, persisting events makes it possible to let users deactivate the gamification module for a certain period without risking the loss of all rewards for the work they have performed in this time span.

Figure 7.2 provides a conceptual illustration of the event tracking facilities incorporated into the HBM, including the most important components of the implementation and their interrelationships. It can be seen that the event tracking process is separated into client-side and server-side activities, and that event objects depicted as small squares serve as communication objects. The most important event tracking steps represented in the figure by numbered circles are as follows:

① **User activity**. Event tracking is activated when the user starts the HBM. Any subsequent interaction of the user with the *user interface* of the software can potentially fire one or more events. As the HBM has no facilities for globally observing the interaction of its users with the interface, appropriate source code must be written for all events that should be monitored, thus making comprehensively tracking all possible events an endeavor that requires significant initial effort. For that reason, the current implementation only tracks those events deemed most relevant for gamification, which includes events related to the user, the workspace explorer, and modeling Petri nets. A full list of the events that are currently tracked in the HBM is provided in the appendix in Section B.1.

② **Event objects**. User actions cause the creation of `UserActionEvent` objects that are posted to an *event notifier* service. The type of the event denotes the concrete action that has been performed by the user and is represented in Figure 7.2 by differently-colored squares. Besides the user interface, events could also be generated by other parts of the application as indicated by the second incoming arrow of the event notifier. In its current implementation, the event notifier works asynchronously and does not report any event handling results; however, this behavior can easily be replaced.

③ **Event handling (client)**. After receiving `UserActionEvent` objects, the event notifier distributes them to any handler class that has subscribed to events of that class. Besides the *user action tracker*, further handlers could also exist as indicated by the second outgoing arrow of the event notifier. The user action tracker is responsible for collecting user actions and transmitting them to the server. Furthermore, it performs certain cleaning activities to ensure, for instance, that no modeling events are sent to the server if the user discards them by closing the model without saving any changes.

④ **Transmission**. After a certain amount of time has passed (e. g., a minute) or a specific number of events (e. g., ten) has been buffered locally, the actual transmission of events to the server takes place. Following already-established practices of the Horus architecture, this is done by serializing events to XML and sending them to the server via a call to a remote procedure that allows for the execution of custom functionality.

Table 7.1: Structure of the HORUS_EVENTS database table.

Column	Data type	Additional details
ID	NUMBER(32, 0)	Primary key
SOURCE	VARCHAR2(500)	Subject having caused the event
EVENT_TYPE	VARCHAR2(500)	Type of event that has occurred
TARGET	VARCHAR2(500)	Object affected by the event
WHEN	DATE	Time stamp of the event
EVENT_XML	XMLTYPE	Arbitrary user data of the event
EVENT_LEVEL	NUMBER(32, 0)	Severity/importance of the event

⑤ **Event handling (server)**. On the server-side, events are received by an *event handler service* that is responsible for deserializing event objects (i. e., transforming them back from XML into Java objects) and delivering them to their respective handlers. Furthermore, this service manages a pool of background threads so that event handling can be carried out in a thread-safe manner that does not interfere with the normal operation of the Horus server. Events may be handled by an arbitrary number of *event handlers* that can either accept all events or restrict themselves to a subset with specific types of event, source, and target. This distinction is highlighted in Figure 7.2 by the difference between the *persistence handler*, which stores event objects in the database and thus has no restrictions, and the remaining three event handlers, each of which only handles one particular type of events.

⑥ **Persistence**. In the final step, events are stored in the database to enable later analyses as previously described. To that extent, a new database table named HORUS_EVENTS was created (see Table 7.1). Besides an artificial primary key, the following information is stored about an event: its *source* (originator of the event, which is most commonly but not necessarily an HBM user), the *type* of event (i. e., the concrete type of user action that was performed), the *target* (object affected by the event, which may for instance be a model, the user himself, or another user), the *date and time* of the event occurrence, and the *level* of the event indicating its severity or importance. Furthermore, each type of event

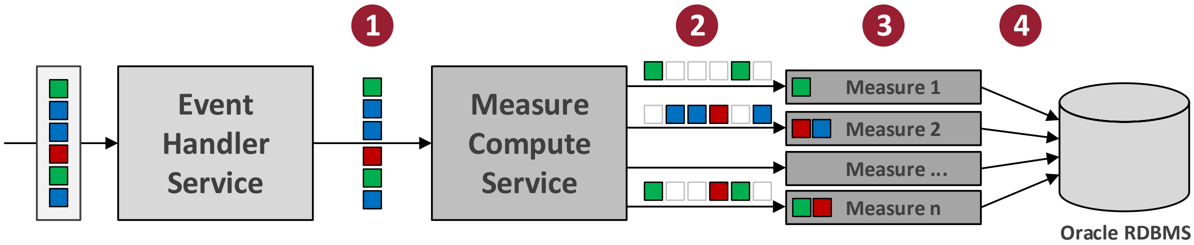


Figure 7.3: Implementation: Standard measure computation process.

may further have arbitrary *use data* that is stored as XML. For instance, for events related to renaming model elements, this field can be used to store the old name so that this information is effectively historicized.

7.2.2 Measures

Measures are performance figures that quantify the behavior of users while working with the HBM and serve as the technical foundation of the reward system described in Section 7.4. As such, they function as the interface between general-purpose event tracking as outlined in the previous section and the core gamification functionality. However, the implementation is generic and could thus also be used for other purposes such as workspace-specific performance dashboards. Sample measures for a user include the account age in days, the number of logins, the overall number of created models, the number of gamification-related points received, and the completeness level of the user profile. For a full list of measures provided by the current implementation, the reader is referred to Section B.2 in the appendix. The process of computing measures directly builds on the event tracking capabilities described in the previous section and is illustrated in Figure 7.3. The most important measure computation steps represented in this figure by numbered circles are as follows:

① **Event handling (server)**. As described in Section 7.2.1, events are transferred from the client to the server, where they are received by the event handler service, which in turn forwards them to appropriate event handlers. The *measure compute service* is one of those handlers, and, identically to the persistence handler shown in Figure 7.2, accepts all events without any restrictions. Its task lies in keeping a list of all measures and distributing incoming events to measures that should be recomputed.

② **Distribution**. In this step, the measure compute service forwards incoming event objects to all affected measures, thereby causing their recomputation. To allow the measure compute service to carry out its task, each measure must define one or more *triggers*, i. e., types of events that should cause an update of the latter. This greatly reduces the computational effort of this step by limiting the number of checks that must be made when a new event arrives. For instance, in Figure 7.3 it can be seen that "measure 1" has defined a single event type as its trigger (indicated by the green square), and will thus not receive events of the other two types.

③ **Computation**. Having been triggered by an incoming event, in this step all affected measures will update their values accordingly. To that extent, each measure requires an *implementation* that provides a recomputation method taking the current value of the measure and an event object as its inputs and delivering the new value of the measure as its output. This computation can be arbitrarily complex and may range from simply increasing a counter by one to performing some calculation with the XML user data associated with the event. A sample implementation of a measure belonging to the latter category is provided in Section B.3 of the appendix.

④ **Persistence**. In the final step, any changes made to the values of measures are written back to the database. To that extent, a new database table named HORUS_MEASURES was created (see Table 7.2). Besides an artificial primary key, the following information is stored about measures: the *id of the user* for whom the measure is stored, the *name* of the measure (i. e., its unique identifier), the *value* that is currently stored for the measure, and the date and time when the measure was last *updated*.

The process as described above applies to standard measures with one or more triggers and an individual implementation. However, besides the standard type, there are also three additional kinds of measures that differ in their methods of computation and in whether they are stored in the database. A full list of these measures can also be found in the appendix in Section B.2. The three additional measure types are as follows:

Table 7.2: Structure of the HORUS_MEASURES database table.

Column	Data type	Additional details
ID	NUMBER(32, 0)	Primary key
USER_ID	NUMBER(32, 0)	ID of the user
MEASURE_NAME	VARCHAR2(500)	Unique identifier of the measure
MEASURE_VALUE	NUMBER(32, 0)	Current value of the measure
UPDATED	DATE	Last update of the measure

- **Global measures** provide a single, global implementation that must be parameterized with a concrete event type when it is used. For instance, the measure *measures.count* provides generic functionality that allows tracking the number of occurrences of any type of event available in Horus. This has the main advantage that instead of requiring a separate implementation for each event type, a single implementation can perform the task for all of them. Unlike standard measures, global measures are only computed and stored in the database if they are actually used by a consumer in conjunction with a particular event type. For example, *measures.count* is only computed and stored for the event type *events.projectManager.modelCreated* (thereby creating the compound measure *measures.count.events.projectManager.modelCreated*), if this particular combination is part of, e. g., the reward condition of a concrete badge.

- **Quality measures** are similar to global measures, but are parameterized with a quality algorithm (see Section 7.2.3) rather than an event type. By using quality measures, it is possible to determine, e. g., the highest absolute value, the highest improvement, or the total sum of improvements a user has ever achieved for a particular quality measure. Analogously to global measures, a quality measure is only persisted in the database if it is actually used.

- **Calculated measures** are based on *expressions* describing how their values should be computed. In the current implementation, these expressions may consist of basic arithmetic operations (*addition, subtraction,* *multiplication, addition*), aggregation functions (*minimum, maximum, average*), *constant* values, and the values of other *measures* (including standard measures, global measures, quality measures, and other calculated measures). Calculated measures are not stored in the database and only computed when the value of an associated measure is updated.

Independently of their type, all measures are defined by adding to a new extension point created for this particular purpose. Creating a new measure requires the specification of a measure type, a unique identifier, a name, a Java class implementing the measure, and a recomputation directive. The latter differs according to the type of the measure: standard measures require one or more triggering types of events, calculated measures require a calculation expression, and the remaining two event types require no such information as they are handled individually.

7.2.3 Quality Framework

To enable the gamification concept described in Section 6.1, it is essential that a system for the evaluation of model quality is in place. This allows providing modelers with real-time feedback regarding the quality of their work and awarding them with points for any quality improvements that they achieve. As such functionality was not required in the HBM before, it was implemented as part of the technical foundations for the gamification module based on the requirements for conceptual model quality frameworks established by Moody [Moo05] and outlined in Section 2.6.2.

The overall **architecture** of the implemented quality framework follows the specifications made in Section 2.6 and is depicted in Figure 7.4. It can be seen that the elements of the framework are arranged in four hierarchical layers. The most important components are *quality metrics*, which are situated on the third level from the top. Each quality metric is an indicator for a specific aspect of the quality of a model and its value falls within the interval $[0; 1]$, with 1 indicating perfection, and 0 denoting absolute faultiness. To avoid semantic ambiguity about their meanings, all metrics require a *measurement procedure* that clearly defines how they should be calculated for a given input model. Going one level up, quality metrics are combined to higher-level constructs called *quality characteristics*. This is done by using the weights w_{ij} to determine

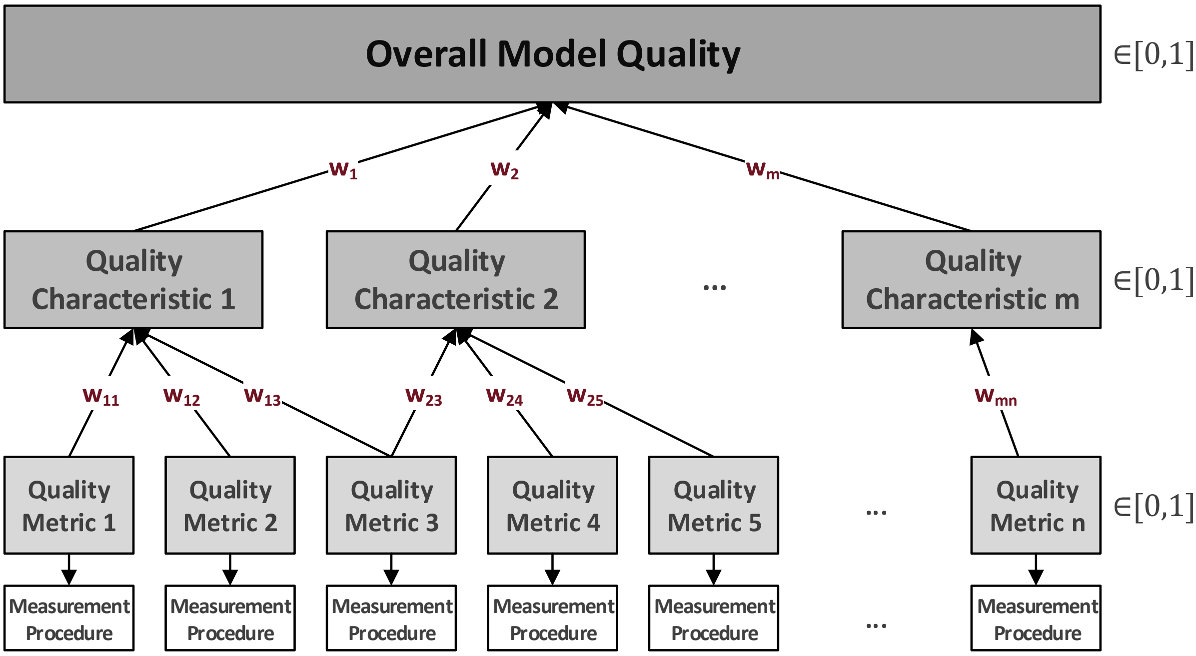


Figure 7.4: Conceptual illustration of the implemented quality framework.

the weighted average of a subset of metrics, so that the result once again falls into $[0;1]$. Conceptually, a quality characteristic summarizes metrics that measure different aspects of the same property, such as readability or aesthetics. Finally, on the top level, the overall quality of a model is computed as the weighted average of all characteristics, wherein the weights $w_1, w_2, \cdots, w_m$ determine the relative importance of each characteristic. Once again, the result is $\in [0;1]$, with 1 representing a model that has perfect quality with regard to the available metrics. The described framework differs from the quality framework principles proposed by Moody [Moo05] in two aspects: Firstly, it omits the intermediate layer of quality sub-characteristics situated between quality metrics and quality characteristics for reasons of structural simplicity. Secondly, it allows for multiple assignments of quality metrics to characteristics, which Moody neither explicitly considers nor prohibits.

At the time of writing, the **set of quality metrics** that has been implemented in the HBM is comprised of the 32 metrics selected in Section 6.1.1 that are assigned to three characteristics: *readability* (7 metrics), *completeness* (14 metrics), and *understandability* (11 metrics). Other model aspects, such as semantic quality and social quality, have not been addressed so far due to their comparable realization difficulty. Altogether, the following properties mirroring the measurement details provided for all metrics in Section 2.7 play an important role for the measurement procedure of a metric:

Absolute measurement Non-standardized value that a metric assumes for a particular model. For instance, the value of the *node overlaps* metric is simply determined as the number of nodes that are involved in overlaps at a given point in time.

Optimization goal Indicates whether optimality for a quality metric can be reached by either minimizing or maximizing its value. For instance, the goal for node overlaps is minimization, whereas the goal for any metric belonging to the *completeness* characteristic is always maximization.

Lower bound Minimum value that a metric can assume given a particular model. For instance, given an appropriately-dimensioned drawing canvas, it can be assumed that the lower bound for node overlaps is always zero.

Upper bound Maximum value that a metric can assume given a particular model. For instance, given a model with n visual elements, the upper bound for node overlaps is n as well, which occurs if every element of the model overlaps with at least one other element.

Quality Standardized value that a metric assumes for a particular model. The quality incorporates value, optimization goal, lower bound, and upper bound to calculate a relative score between zero and one for a model at a given point in time.

All measurement procedures are *generic* in the sense that they do not make any reference to a particular type of model available in the HBM. To that extent, *graphs* are used as an additional layer of abstraction between the original source models (e. g., process models, object models, organizational models) and measurement procedures. Accordingly, any kind of model first has to be transformed into a graph by an appropriate *transformer* class. This characteristic of the implementation directly mirrors the use of business process graphs as a canonical form of business process models as practiced from Section 2.5

onwards. From a technical point of view, this approach is required because the different kinds of models that are used in Horus do not share a common data structure and interface, thereby preventing the realization of a single measurement procedure that is compatible with all model types. While generality may not be necessary for all metrics (i. e., some may be highly specific and only defined for one particular type of model), it greatly facilitates realizing metrics that work regardlessly of the model type, such as node overlaps, where it is of little importance what kinds of nodes overlap as long as they occupy visual space. To illustrate this point, the implementation of this measure is provided in Section B.4 of the appendix.

Eventually, the decision which model types a specific metric shall support must be made during the implementation of its measurement procedure. For that purpose, besides providing an appropriate implementation, the programmer must explicitly specify the model types that a metric is compatible with. Consequently, of the set of implemented metrics, only a (potentially empty) subset may be valid at any given time depending on the type of the current model. At the present time, the 32 implemented metrics mentioned above all exclusively support Petri net process models and process architectures. The set of available metrics can be narrowed down even further by means of a manual configuration on the level of a repository or individual workspaces. To that extent, the configuration dialog shown in Figure 7.5 may be used by users with administration rights for the target entity.

The **runtime dynamics** of the quality framework are illustrated by the Petri net shown in Figure 7.6. The first step after the user has opened a new model consists of loading all valid quality metrics. To that extent, it is first checked whether an active metrics configuration can be found for the enclosing workspace of the model, for any of its parent workspaces, or lastly for the repository itself. Should this be the case, the subset of metrics activated in the configuration serves as the starting point; otherwise, the set of all implemented metrics is used. Next, the set of metrics is narrowed down to those that support the type of the current model. In the second step, the input model is converted into its canonical, graph-based representation using an appropriate model transformation class. In the third step, the current values of all metrics are computed using the model graph as an input. Afterwards, the results of the previous step are visualized in the user interface, thus effectively turning

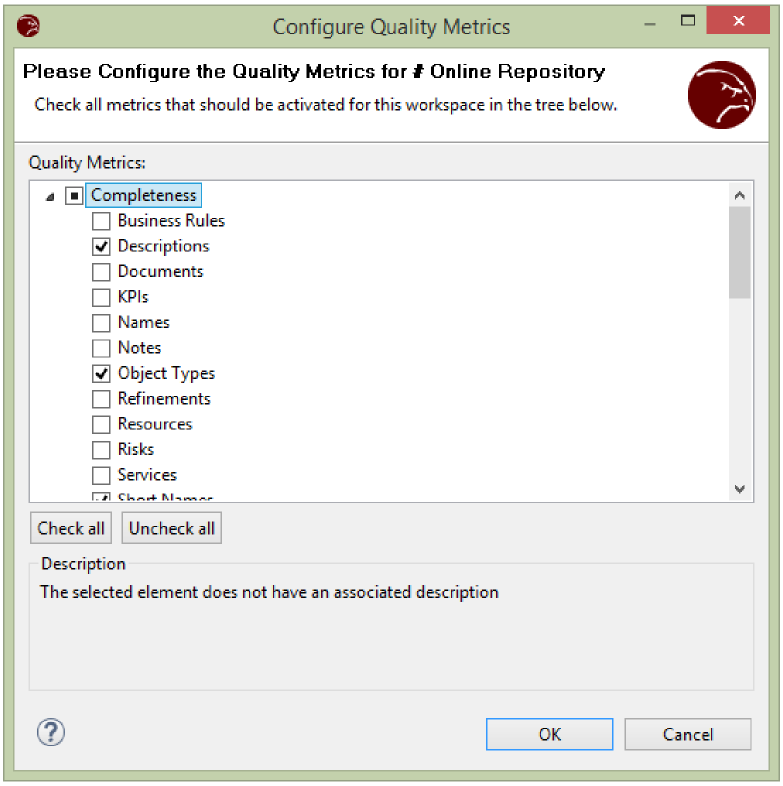


Figure 7.5: Implementation: Quality metrics configuration.

the quality metrics into actionable information that modelers can employ to improve the quality of their models. As the two activities of transformation and computation may result in considerable computational effort for large models with many active quality metrics, they are performed asynchronously (i. e., in the background and without blocking the user interface) and are only carried out when the user has stopped making changes to the model. In the final step, the quality framework waits for the next input by the user, which may fall into three general categories: the user resumes changing the contents of the model, thereby triggering a new iteration of the transformation-computation-visualization loop; the user saves the changes he made, whereupon an event is fired that may be consumed by any class interested in model quality changes; the user closes the model, thereby terminating the process.

Just as measures, quality metrics are defined by adding to a new extension point created for this particular purpose. Creating a new quality metric requires the specification of a unique identifier, a metric name, an optimization goal,

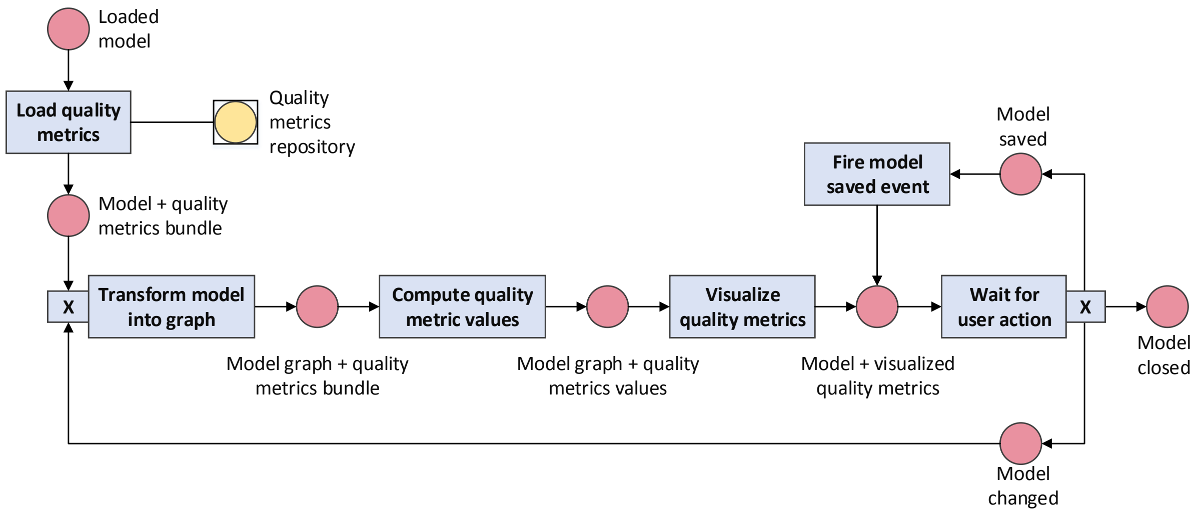


Figure 7.6: Implementation: Quality framework runtime behavior.

a Java class implementing the metric, and a list of model types for which the metric is valid. Using a second extension point, developers can then add metrics to quality characteristics while specifying weights for the computation of a weighted arithmetic mean on all non-leaf levels of the quality hierarchy.

7.3 Gamification Architecture

As previously mentioned, users working with the Horus Business Modeler can work with three different types of repositories: local repositories, database repositories, and server repositories. As only the latter allows for the execution of server-side business logic, which is an obligatory requirement for gamified business process modeling, gamification is only available for server repositories, and thus the remaining two options will be disregarded in the following. Consequently, users who are switching between repositories of different types during a single modeling session may find gamification functionality to alternate between being enabled and disabled. Similarly, it should be noted that all data maintained in the context of gamification is server-specific. This means that unlocking a badge or earning 100 points on one server does not automatically yield the same rewards on any other connection. Thus, users working on multiple modeling projects may find their gamification status to vary across repositories.

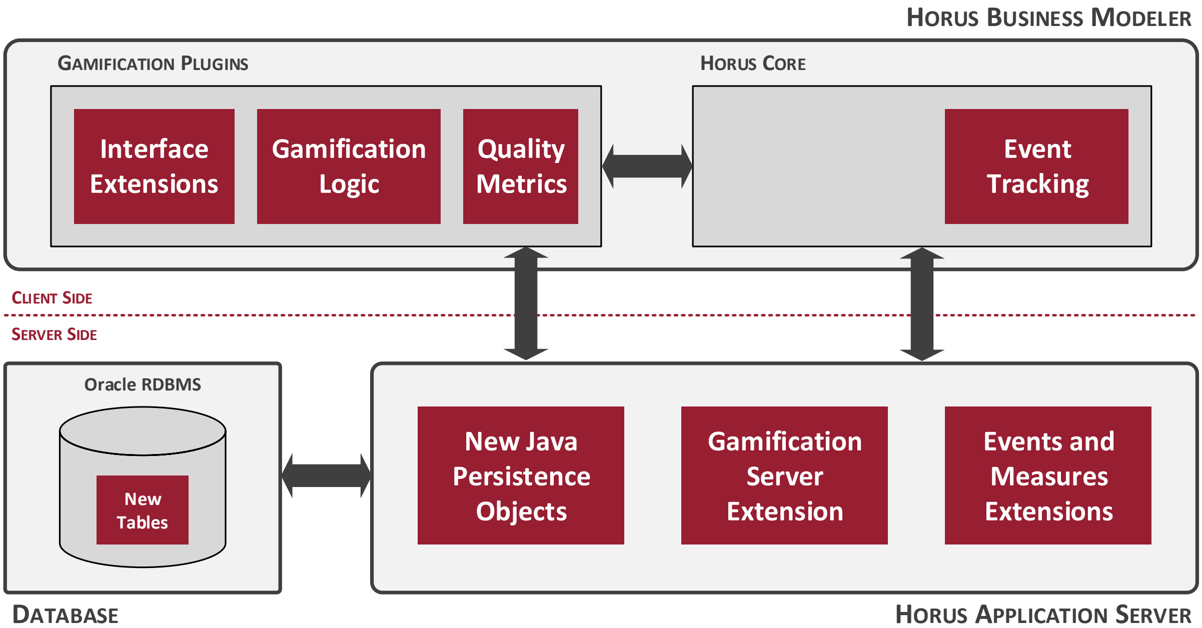


Figure 7.7: Implementation: Architecture overview.

A high-level illustration of the architecture of Horus is provided in Figure 7.7. Herein, red boxes denote elements that have been added in the context of the gamification module, and further details have been omitted in the interest of brevity and clarity. Focusing on the core of the HBM, the only notable extension can be found in the event tracking capabilities described in Section 7.2.1. Any additional functionality is provided in separate gamification plugins; this includes the quality framework outlined in Section 7.2.3, the entire business logic relating to gamification, and extensions to the interface of the Horus client through which gamification-functionality is presented to the user. On the server-side, extensions for event handling and persistence, and the computation of performance measures as described in Section 7.2.2 are provided as a complement to client-side event tracking. Furthermore, a gamification server extension encapsulates all functionality related to gamification, such as managing a list of active users and their gamification data, checking whether an incoming event results in a user satisfying the conditions for obtaining a reward, and sending notifications about new rewards to clients. Lastly, the database used by the Horus server was extended with new tables as mentioned above, which possess corresponding Java classes for object-relational

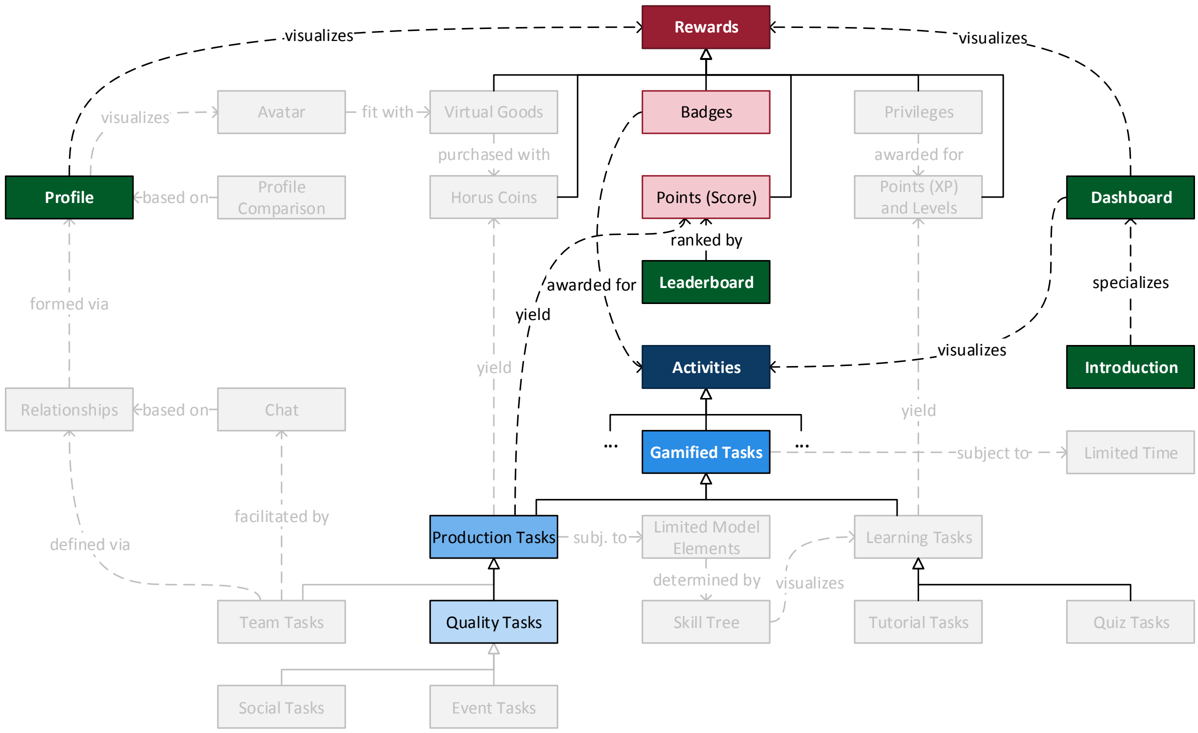


Figure 7.8: Gamification design lens: Implemented subset of elements.

mapping on the application server. In summary, the architecture of Horus and the distribution of gamification functionality across server and client allows for a distribution of the computational workload and ensures that it is difficult for clients to claim rewards without having actually performed the required actions.

7.4 Gamification Features

Having outlined the general architecture of the gamification module, this section provides a detailed description of the implemented features relating to gamification. These features are as follows: a gamification status panel (Section 7.4.1), user profiles (Section 7.4.2), points (Section 7.4.3), badges (Section 7.4.4), a leaderboard (Section 7.4.5), and an introduction into the module (Section 7.4.6). Where appropriate, references back to the technical foundations presented in Section 7.2 will be made.

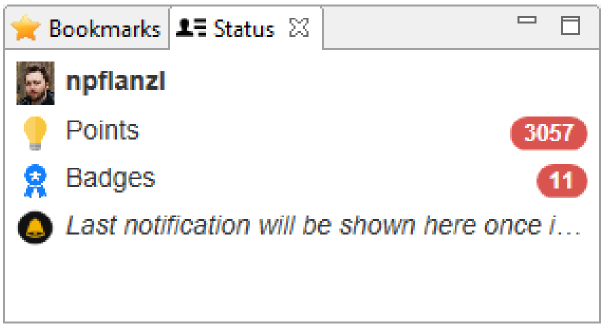


Figure 7.9: Implementation: User status panel with points and badges.

Due to the extent of the gamification design concept presented in Chapter 6 and the time and resource constraints that this research project was subject to, only the subset of features depicted in Figure 7.8 could be implemented. This subset instantiates the Points, Badges, Leaderboards (PBL) blueprint of gamification for process modeling by means of quality tasks and provides additional functionality such as a profile and a dashboard. Overall, the fraction of elements contained in Figure 6.1 that were implemented as shown in Figure 7.8 is 39%. Furthermore, the gamification module addresses all of the concept lenses examined in Section 6.2 to some extent, with the coverage being 64% for the model quality lens, 57% for the training lens, 56% for the extrinsic motivation lens, 53% for the flow lens, and 40% for the self-determination lens. However, it should be noted that this does not necessarily indicate the capability of the implementation to satisfy the goals associated with the respective lenses. For instance, despite the fact that 57% of the training lens is covered, the most important features—tutorial tasks, quiz tasks, and the skill tree—are still missing.

7.4.1 Status

The status panel shown in Figure 7.9 serves as a brief summary of all gamification-related achievements the user has earned on a given repository, and is displayed in the lower-left corner of the HBM by default. To that extent, it displays the account name, a small thumbnail of the user's profile picture, and, more importantly, the number of points and badges the user has received. As users may be working in multiple contexts (i. e., on multiple repositories) during a single session in the HBM, the status panel may not be shown at all times,

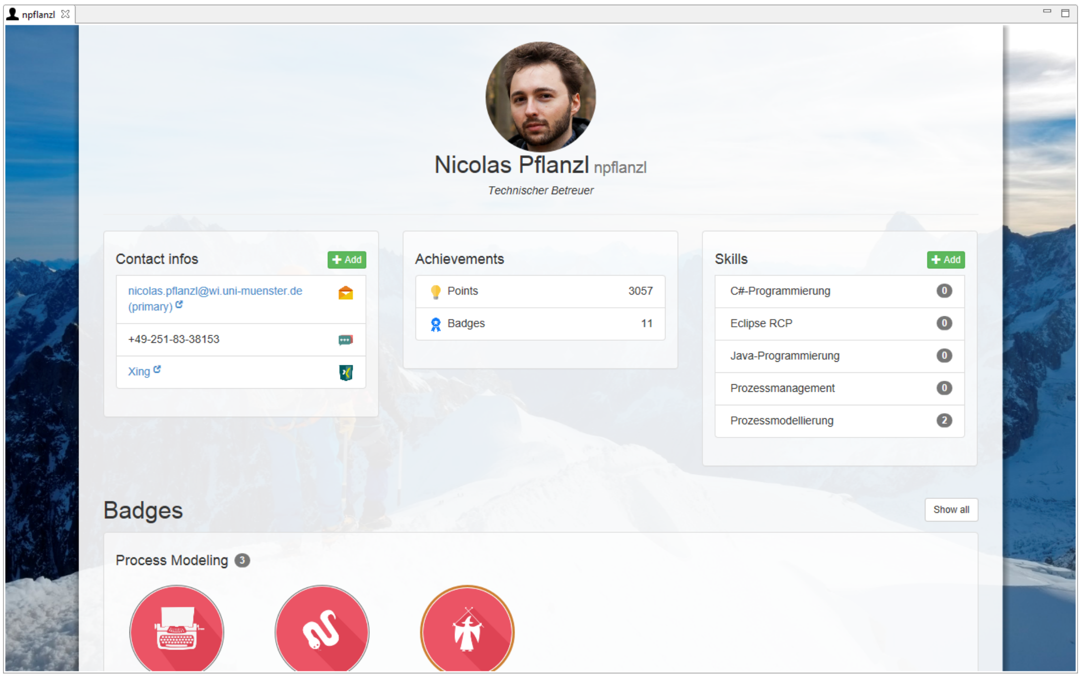


Figure 7.10: Implementation: User profile with partial badges list.

but only in appropriate contexts where gamification is enabled. Accordingly, when the context changes, it is either updated with relevant information, or hidden. Thus, whether the status panel is visible or not serves as a simple indication for whether the user is currently working on a repository supporting gamification. Besides this informatory function, the status panel also serves as a *notification area* in which messages are displayed when the user obtains points or unlocks a badge. Notification examples are shown in the following, respective subsections.

7.4.2 Profile

While the non-gamified implementation of the Horus Business Modeler (HBM) already allows for the specification of basic user data (first name, last name, email address, description, and a profile picture), this information is only visible in administration dialogs and not made available in any public views. Thus, one primary concern of the gamification module was an extension of the set of user data that can be specified as well as the provision of a view that actually displays the entirety of this data to modelers. To that extent, the user profile window shown in Figure 7.10 was implemented. The top part of this view depicts the aforementioned basic user information including the user's account name. When viewing their own profile, users may click the profile picture to quickly access the configuration dialog for basic user data. The middle area of the profile displays *contact information*, such as email addresses, telephone numbers, and Skype, Xing, or Facebook accounts, and *skills* that users claim to possess. Once again, when viewing their own profiles, users can find appropriate facilities for editing this data. In turn, when viewing the profiles of others, users may choose to "endorse" a skill, i. e., support the respective user's claim to possess said ability. The number of current endorsements for each skill is always indicated on the right-hand side of the respective entry. Finally, in the center of the middle area, a brief record of gamification-related achievements indicating the current number of points and badges of the user is provided. Lastly, the bottom part of the profile lists the badges held by the user as described in Section 7.4.4. With the introduction of additional gamification-related functionality, the user profile may further be extended to include information such as the tutorials already completed by the user or his current experience level (see Section 6.1).

7.4.3 Points

Points are one of the two main game design elements of the gamification module (the other being badges), and are thus connected to all parts of its implementation. For instance, they are displayed in both the status panel and the profile and form the basis for the leaderboard. However, they are most prominent in the Petri net model editor, where they can occupy a considerable portion of the available screen real estate should the user decide to display all visual elements simultaneously. This is illustrated in Figure 7.11 in which all user interface elements besides the central modeling canvas and the main menu at the top are part of the gamification module. The functionality related to points can broadly be subdivided into three different areas that will be discussed in the following: the *computation* of a points score and reward for the current model based on the implemented quality framework, providing *real-time quality feedback* to users to enable them to create models of a high quality and earn further points, and a simple *help system* giving users access

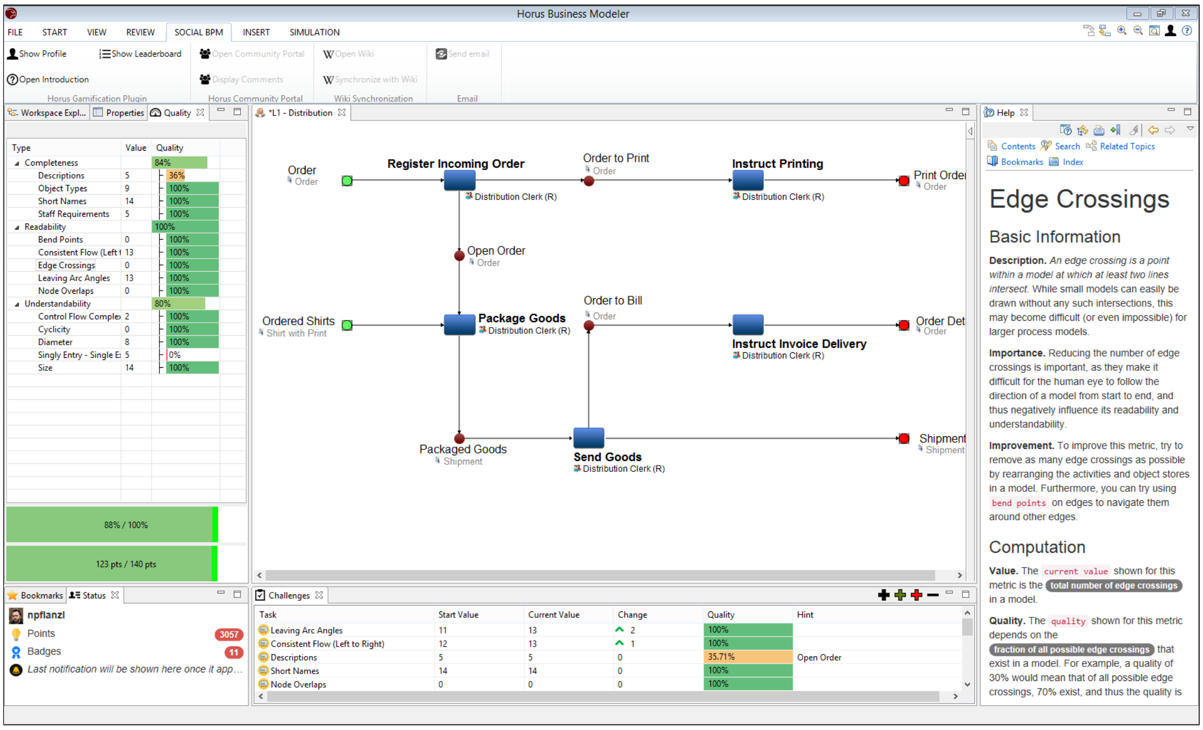


Figure 7.11: Implementation: Modeling view with all interface elements.

to further information about quality metrics and pinpoint particular areas of a model that can be improved.

Computation

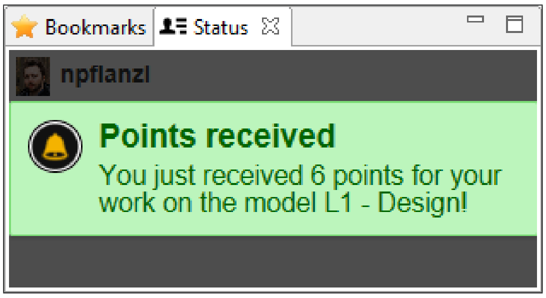
Given two business process graphs G_{t-1} and G_t representing the same process model at two subsequent points in time $t-1$ and t, the set of active quality metrics as represented by their relative measurement procedures R, an experience constant exp, and a scaling constant $scale$, the number of points that a user should receive for changing G_{t-1} to G_t is computed according to Algorithm 1, which defines the following two functions:

- **ComputePoints**: This function computes the points score for a given process graph. If the graph is empty or disconnected, the result is manually set to 0 points. The first is necessary as an empty model cannot contain any quality defects, and would thus receive a perfect evaluation. The latter is required due to the fact that many quality metrics assume connectedness and are unable to provide an accurate result otherwise. If both conditions are fulfilled, the points score of G is determined as

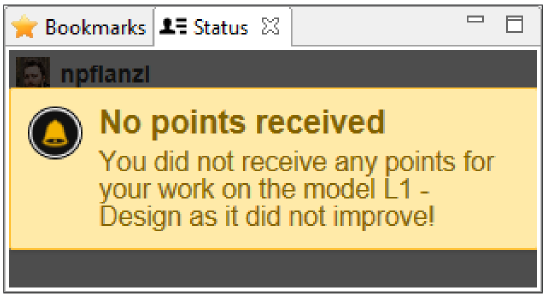
Algorithm 1: Compute points reward for changes to a process model

ComputePoints $(G, R, exp, scale)$
  inputs : Business process graph G; other inputs as above
  output : Points score for G denoted by $points$
  $points \leftarrow 0$;
  if *isEmpty(G)* ***or*** *isNotConnected(G)* **then**
    return $points$;
  foreach *measurement procedure* $r \in R$ **do**
    $points \leftarrow points + r(G) * exp$;
  $points \leftarrow points * \min(1, |N|/scale)$;
  return $points$;

ComputeReward $(G_{t-1}, G_t, R, exp, scale)$
  inputs : Business process graphs G_{t-1}, G_t at time points $t-1, t$; relative measurement procedures R; experience constant exp; scaling constant $scale$
  output : Amount of points to reward denoted by $reward$
  $reward \leftarrow ComputePoints(G_t, R, exp, scale)$;
  $reward \leftarrow reward - ComputePoints(G_{t-1}, R, exp, scale)$;
  return $\max(0, reward)$;



(a) Points received.



(b) No points received.

Figure 7.12: Implementation: Points reward notification.

the sum of its values for every measurement procedure multiplied with an experience constant. In the current implementation, this constant has a value of 10, meaning that the maximum number of points that a model can have is 320 points if all 32 quality metrics are active. Finally, the number of points is scaled in dependence of the size of the model so that all models smaller than the scaling constant are proportionally devaluated. This penalizes very small models, for which maintaining a high quality level is comparably easy, to discourage modelers from creating only trivial models to increase the height of their reward. In the current implementation, this constant has a value of 5.

- **ComputeReward**: By invoking the `ComputePoints` routine for both business process graphs, the increase (or decrease) in points yielded by the changes made by the current user can be determined. If the difference is positive, it is distributed to the modeler as a reward and the user is notified with a pop-up message as shown in Figure 7.12a. Following the recommendation of Schell discussed in Section 6.1.2, a decrease in model points does not result in a penalty, but simply the absence of a reward. This is also communicated to the user with a notification as shown in Figure 7.12b.

The time complexity of Algorithm 1 is directly determined by the relative measurement procedures contained in R so that its overall complexity is equal to the largest complexity of any $r \in R$. Therefore, ensuring that the implementations of all measurement procedures is efficient—if necessary through the

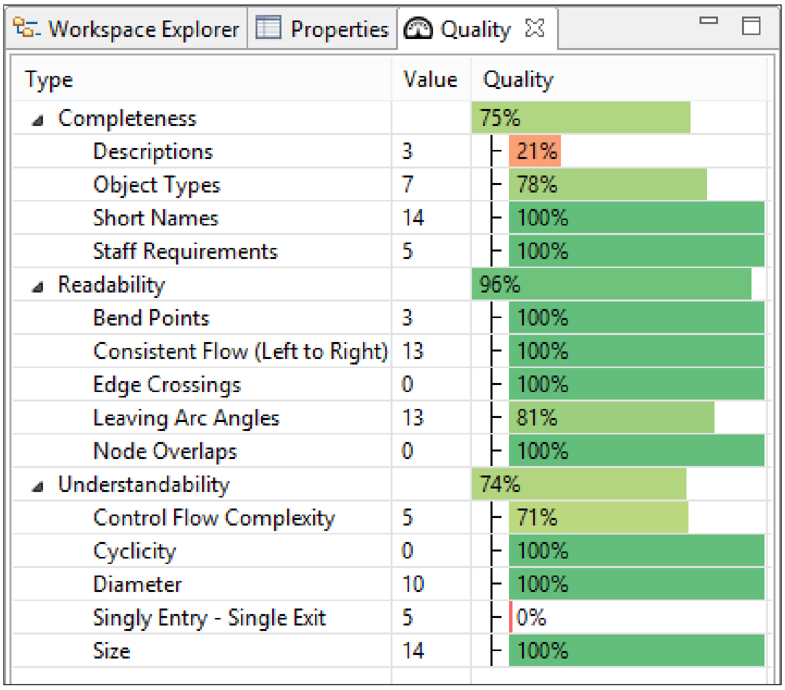


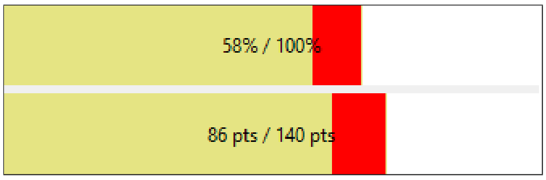
Figure 7.13: Implementation: Quality characteristics and metrics.

use of heuristics rather than deterministic algorithms—is of utmost importance for providing quality feedback in (approximate) real-time. Assuming that the number of edges in most business process graphs is larger than the number of nodes, the metric edge crossings currently has the highest complexity of all implemented metrics, i. e., $O(|A|^2)$, with A denoting the set of arcs.

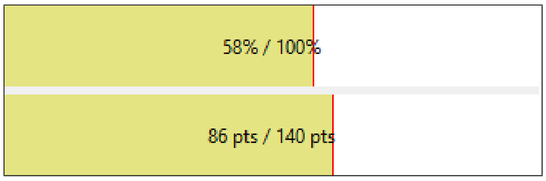
Real-time Feedback

An essential requirement for quality tasks that is also related to the theoretical foundations of gamification such as flow theory and goal-setting theory (cf. Section 3.5) is the availability of real-time quality feedback. The overall interface depicted in Figure 7.11 provides such feedback in multiple places that are described in detail in the following.

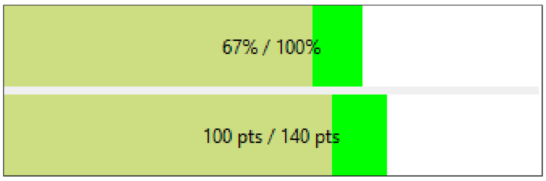
Firstly, the quality panel on the left side of the interface shown in Figure 7.13 contains a hierarchical tree of quality characteristics and all associated quality metrics that are currently active. Due to the possibility for the assignment of metrics to multiple characteristics, some metrics may appear more than once, although this is not the case in Figure 7.13. For each metric, the absolute and relative measurements are shown in the same line, the former being referred



(a) Negative feedback.



(b) Neutral feedback.



(c) Positive feedback.

Figure 7.14: Implementation: Modeling feedback for quality and points.

to as "value", and the latter being converted into a percentage. Furthermore, a color gradient ranging from dark red to dark green is used to enrich the representation of relative measurements with additional visual clues. For quality characteristics, percentages are computed as the weighted average of all associated quality metrics. Finally, all values are updated in real-time whenever the user has made changes to the underlying model.

In addition, modelers are also provided with aggregated feedback about the quality and points scores of the model as shown in Figure 7.14. To that extent, the quality score is computed as the weighted average of all quality characteristics, and the points score is determined according to Algorithm 1. Both indicators are visualized as progress bars that allow users to quickly assess whether they have increased or decreased the quality of the model since the last save operation. This is achieved by coloring the respective portion of the progress bar that has been "lost" or "gained" in red (see Figure 7.14a) or green (see Figure 7.14c), respectively.

To receive more detailed feedback about individual metrics, users can add them to the challenges panel shown in Figure 7.16 that resides at the bottom of the interface. This can be done by means of the black, green, and red "plus" buttons, which allow adding a single metric, all available metrics, or the five "worst" (i. e., with the lowest relative measurements) metrics. Pressing the black button opens the dialog window shown in Figure 7.15, that lets users choose from the set of available metrics, and, upon the selection of an entry,

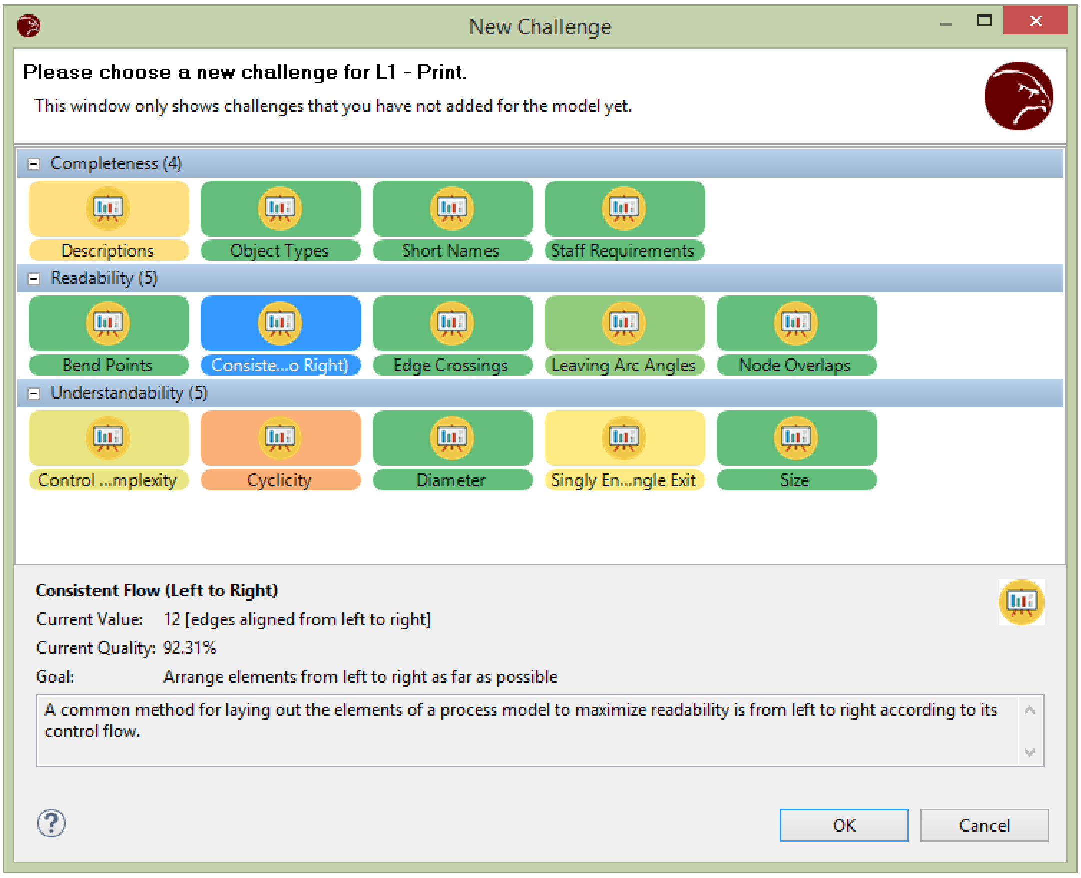
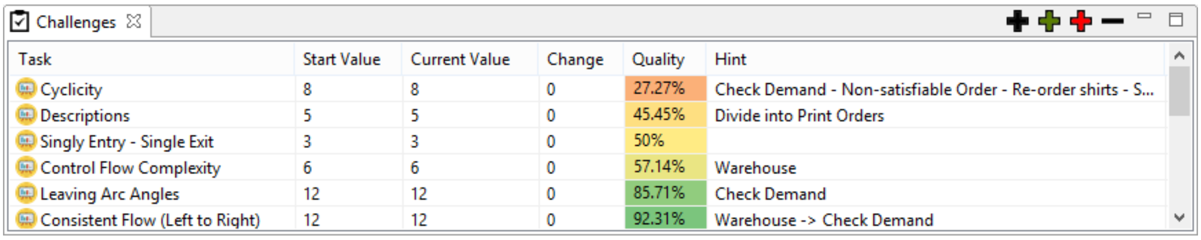


Figure 7.15: Implementation: Modeling challenge creation.

provides them with additional information. As in Figure 7.13, color is used to emphasize the current state of the model with regard to a particular quality concern. In addition to the data shown in the quality panel, the challenges panel also displays the following information: the *start value* of the metric (i. e., the absolute measurement made when the model was last saved), the *change* in absolute measurement since the last save operation, and a *hint* pointing to an element or set of elements where the metric can be improved. Just as all other quality feedback mentioned so far, any changes to the model are reflected in the challenges panel in real-time.

Help

While some quality metrics such as edge crossings or node overlaps are conceptually simple and can thus be easily understood even by novice modelers, other metrics, for instance connector mismatch and control flow complex-



Challenges

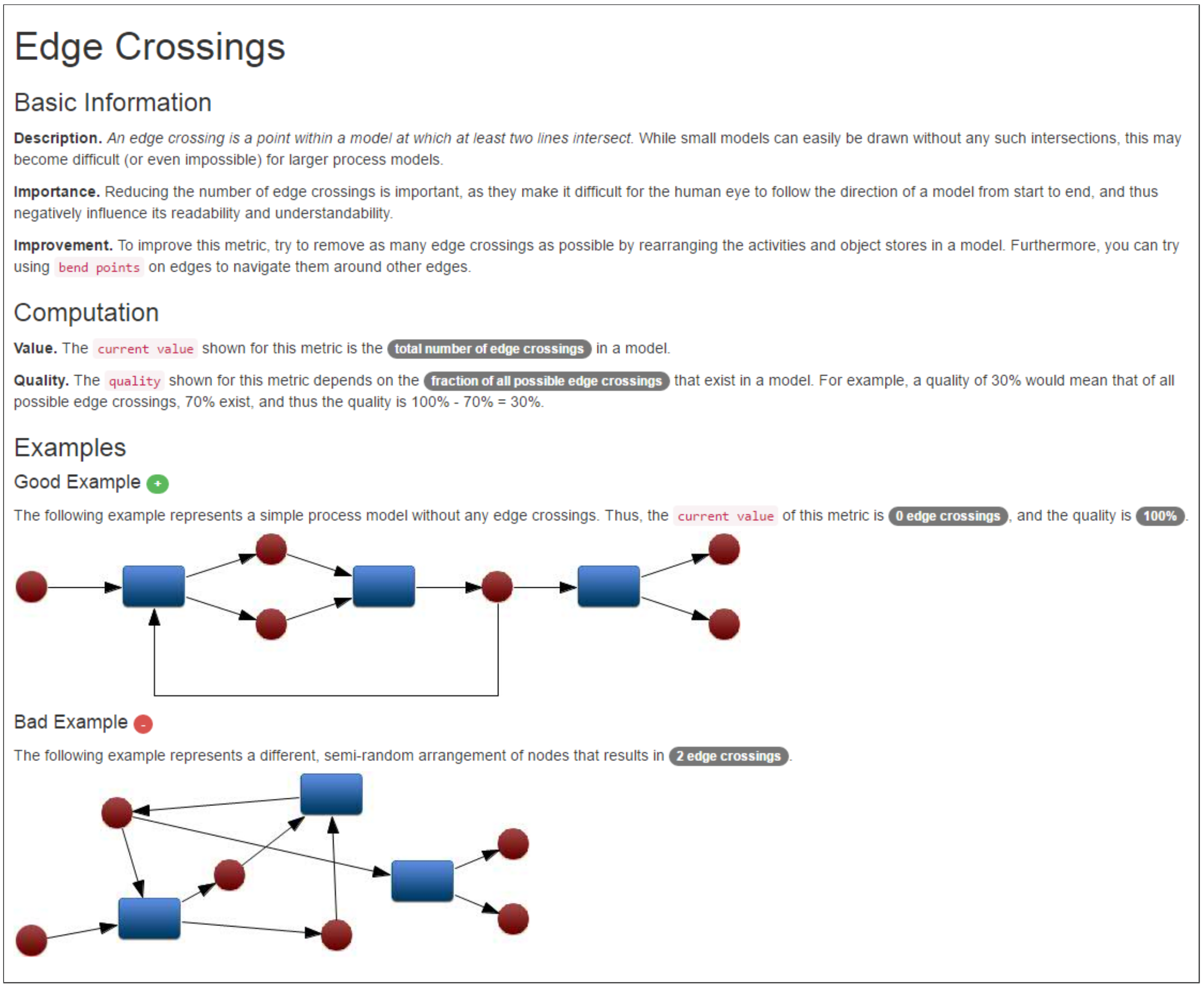
Task	Start Value	Current Value	Change	Quality	Hint
Cyclicity	8	8	0	27.27%	Check Demand - Non-satisfiable Order - Re-order shirts - S...
Descriptions	5	5	0	45.45%	Divide into Print Orders
Singly Entry - Single Exit	3	3	0	50%	
Control Flow Complexity	6	6	0	57.14%	Warehouse
Leaving Arc Angles	12	12	0	85.71%	Check Demand
Consistent Flow (Left to Right)	12	12	0	92.31%	Warehouse -> Check Demand

Figure 7.16: Implementation: Modeling challenge list.

ity, are more difficult to comprehend without prior modeling expertise. To extenuate the complexity of these metrics, a rudimentary help system was integrated into the HBM and is shown on the right-hand side of the interface in Figure 7.11. To access the help, modelers can press F1 on their keyboard upon selecting any metric in either the quality panel, the challenges panel, or the challenges dialog window. In the resulting window (see Figure 7.17), general information about the metric, its importance for model quality, and strategies for its improvement are presented. Furthermore, users are provided with details about the computation of the metric (including its absolute and relative measurement), as well as sample models of high and low quality. Through this information, it is expected that novice users are supported in their capability to operationalize all available quality metrics more effectively.

7.4.4 Badges

The badge system implemented in the HBM directly builds on the technical foundations outlined in Section 7.2: actions performed by users trigger events that cause the recomputation of performance metrics, which in turn may lead to a badge being unlocked. Overall, from an implementation point of view, the life cycle of a badge can be subdivided into three distinct phases. Firstly, in the initial state the badge of interest does not yet exist and has to be *defined*. Secondly, after being defined, users may perform actions towards *unlocking* the badge as designated by its reward criteria. Lastly, after the badge has been unlocked by a particular user, it is *presented* on his user profile as part of his record of achievements. The following three paragraphs provide a more detailed description of these three phases.



Edge Crossings

Basic Information

Description. *An edge crossing is a point within a model at which at least two lines intersect.* While small models can easily be drawn without any such intersections, this may become difficult (or even impossible) for larger process models.

Importance. Reducing the number of edge crossings is important, as they make it difficult for the human eye to follow the direction of a model from start to end, and thus negatively influence its readability and understandability.

Improvement. To improve this metric, try to remove as many edge crossings as possible by rearranging the activities and object stores in a model. Furthermore, you can try using `bend points` on edges to navigate them around other edges.

Computation

Value. The `current value` shown for this metric is the **total number of edge crossings** in a model.

Quality. The `quality` shown for this metric depends on the **fraction of all possible edge crossings** that exist in a model. For example, a quality of 30% would mean that of all possible edge crossings, 70% exist, and thus the quality is 100% - 70% = 30%.

Examples

Good Example +

The following example represents a simple process model without any edge crossings. Thus, the `current value` of this metric is **0 edge crossings**, and the quality is **100%**.

Bad Example -

The following example represents a different, semi-random arrangement of nodes that results in **2 edge crossings**.

Figure 7.17: Implementation: Help page details.

Definition

The definition of a badge consists of all such activities that make it known to the system and thus enable users to unlock the former. While it is theoretically possible to hard-code all rewards directly into the source code of the Horus Business Modeler (HBM), this may be considered a software development "anti pattern" [BMMM98] leading to a significant overhead and requiring source code modifications whenever changes to the set of available badges should be made. Thus, the implemented badge system makes use of the *extension point* mechanism of the Eclipse RCP described in Section 7.1. Consequently, the definition of badges is carried out by writing appropriate XML snippets or using the graphical facilities provided by the Eclipse IDE as a more user-friendly alternative. Following [HE11], a new badge is created by first defining its *signifier* consisting of the name of the badge, its description, and a visual. Special consideration must be given to the wording of the description as it is the only detailed information indicating to users how a badge may be unlocked. Next, the *level* of the badge must be specified. This can only be done implicitly by referencing an optional predecessor badge. Consequently, the level of a badge can be determined as the total number of predecessors that it possesses. Finally, the *completion logic* of the badge must be described to clarify the requirements that a user must satisfy to unlock a certain badge. The extension points allows for the declaration of multi-level reward criteria that can be nested to an arbitrary depth and connected with basic Boolean operators.

Listing 7.1 provides a comprehensive illustration of the possibilities for rewarding badges by using the extended Backus-Naur form (EBNF) to define a simple completion logic language. EBNF was chosen to create a more concise version of the much more verbose but equivalent representation as an XML schema. On the top level, the specification consists of a single *expression*, which may be terminal or nonterminal. *Terminal expressions* are built using the measures described in Section 7.2.2 as a foundation and yield a Boolean value as a result. Overall, there are four types of terminal nodes: *measure expressions* (comparison of a measure with a static value), *measure comparisons* (comparison of two measures), *event expressions* (comparison of a static value with a global measure parameterized with an event type) and *quality expressions* (comparison of a static value with a quality measure parameterized with a quality metric). All four types may make use of common *relational operators*

```
CompletionLogic = Expression;

Expression = TerminalExpression | NonterminalExpression;
TerminalExpression = MeasureExpression | MeasureComparison | EventExpression | QualityExpression;
NonterminalExpression = And | Or | Not;

(* terminal expressions *)
MeasureExpression = "measureExpr(", Measure, Operator, Value, ")";
MeasureComparison = "measureComp(", Measure, Operator, Measure, ")";
EventExpression = "eventExpr(", GMeasure, EventType, Operator, Value, ")";
QualityExpression = "qualityExpr(", QMeasure, Metric, Operator, Value, ")";

(* nonterminal expressions *)
Or = "or(", { Expression }+, ")";
And = "and(", { Expression }+, ")";
Not = "not(", Expression, ")";

(* Boolean operators *)
Operator = '<' | '<=' | '=' | '!=' | '>=' | '>';

(* pre-defined sets of symbols *)
Value = ? set of rational numbers ?;
Measure = ? all available standard & calculated measures ?;
GMeasure = ? all available global measures ?;
QMeasure = ? all available quality measures ?;
EventType = ? all available event types ?;
Metric = ? all available quality metrics ?;
```

Listing 7.1: Simple language for the completion logic of badges as an extended Backus-Naur form (EBNF).

such as “greater than”, “equal” or “not equal”. *Nonterminal expressions* in turn are represented by the three Boolean operators *conjunction* (AND), *disjunction* (OR), and *negation* (NOT). Whereas the first two may contain an arbitrary, positive number of both terminal and nonterminal sub-expressions, the latter consists of exactly one such item. This allows for the hierarchical definition of complex, interlinked completion criteria. Put into action, criteria such as those of the examples presented in Table 7.3 can be specified. Whereas the left-hand side of the table describes the completion logic using the language defined in Listing 7.1, the right-hand side depicts the equivalent but longer XML-based representation. Furthermore, while the first three badges employ a terminal expression already at the top level, the fourth badge uses a non-terminal conjunction to combine two second-level terminal expressions.

Table 7.3: Definition of the completion logic of four sample badges.

Based on EBNF in Listing 7.1	XML

Babbler (Level 1)

```
measureExpr(measures.modeling.text
    .charactersWritten >= 10000)
```

```
<measureExpression
    measure="measures.modeling.text.
        charactersWritten"
    operator="greater than or equal"
    value="10000">
</measureExpression>
```

Power User (Level 2)

```
eventExpr(events.user.login
    measures.count >= 100)
```

```
<eventExpression
    eventType="events.user.login"
    measure="measures.count"
    operator="greater than or equal"
    value="100">
</eventExpression>
```

You Shall Not Cross (Level 2)

```
qualityExpr(
    measures.quality.metric.decreaseSum
    qualityMetrics.aesthetics
        .edgeCrossings <= -100)
```

```
<qualityExpression
    measure="measures.quality.metric.
        decreaseSum"
    metric="qualityMetrics.aesthetics.
        edgeCrossings"
    operator="less than or equal"
    value="-100">
</qualityExpression>
```

Amateur Constructivist (Level 1)

```
and(
    measureExpr(
        measures.modeling
            .numModelsDeleted <= 5)
    measureExpr(
        measures.modeling
            .numModelsCreated >= 10))
```

```
<AND>
    <measureExpression
        measure="measures.modeling.
            numModelsDeleted"
        operator="less than or equal"
        value="5">
    </measureExpression>
    <measureExpression
        measure="measures.modeling.
            numModelsCreated"
        operator="greater than or equal"
        value="10">
    </measureExpression>
</AND>
```

Unlocking
Once a badge has been defined, users can perform actions towards fulfilling its reward criteria in order to unlock it. The entire process consists of five separate steps and is illustrated on the conceptual level by means of a concrete example in Figure 7.18. The five steps of badge unlocking are as follows:

① **User activity, event handling, measure computation**. The first step consists of the generic activities outlined in Section 7.2.1 and Section 7.2.2 and is thus not exclusive to the badge system. Therefore, the reader is referred to the respective sections for more detailed information. In the example given in Figure 7.18, a particular user has performed a "blue" event, which has been transmitted to the server, passed to the measure computation service, and triggered updates of the measures M1 and M2.

② **Completion check**. After being triggered and recomputed, measures employ the visitor pattern [GHJV95] to notify all dependent badges that a reexamination of their completion logic should occur. For each measure, the set of badges to notify is comprised of all badges that use the former in any of their terminal expressions and that have not yet been earned by the user in question. Each notified badge reacts by checking its completion logic to investigate whether it has just been unlocked. To that extent, starting with the top-level expression, all potentially existing sub-expressions are recursively resolved until a single Boolean value is obtained that indicates whether the badge should be awarded. If this is the case, the *badge management service* holding a dictionary of all (currently active) users and their badges is informed. In the example at hand, the "blue" event sent to the server triggers updates of the measures M2 and M3. Consequently, their values are recomputed and the badges B1, B2, and B3 notified. As the user already holds badge B2, no further action is required in this instance. However, the expressions of badges B1 and B3 are reevaluated, which results in the former being unlocked.

③ **Persistence**. The fact that the user has unlocked a new badge is persisted in the database. Furthermore, the badge management service updates its local dictionary and can thus answer future requests about the user in question possessing badge B1 correctly without accessing the database.

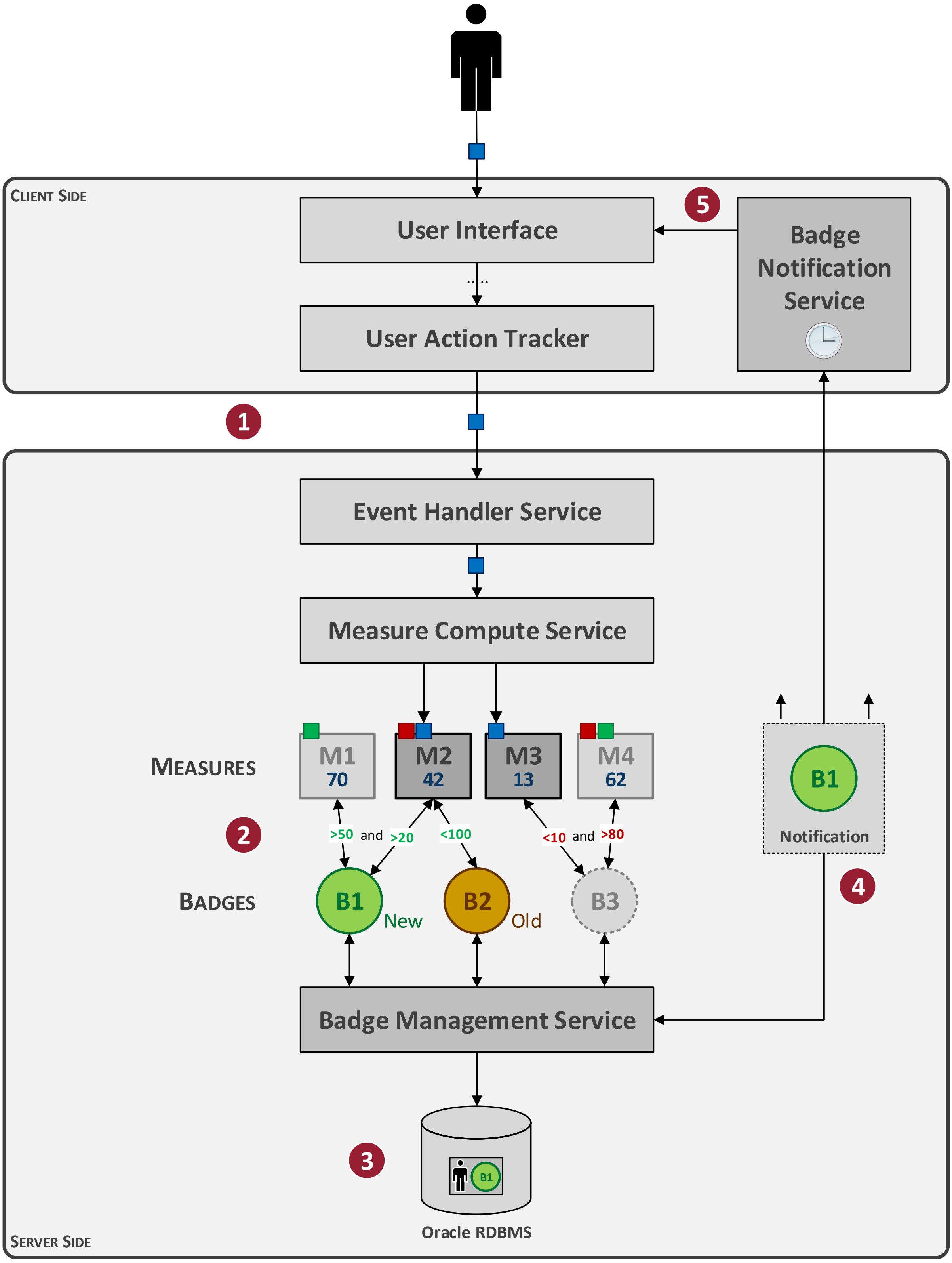


Figure 7.18: Implementation: Conceptual illustration of badge unlocking.

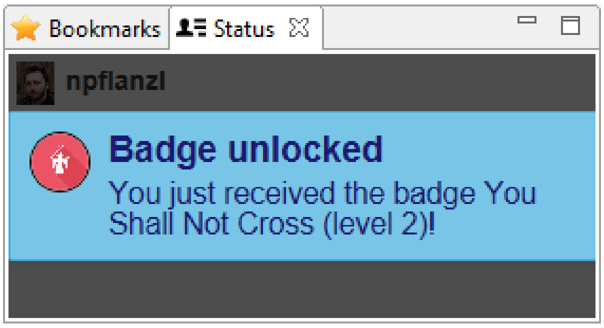


Figure 7.19: Implementation: Badge unlock notification.

④ **Notification delivery**. After unlocking a badge, the badge management service ensures that the affected user is notified of his newly-obtained achievement. For that purpose, a notification with the internal ID of the respective badge is generated and delivered back to the user at the next possible point in time. To that extent, the client executes a *badge notification service* that polls the server for any pending notifications in regular intervals of, e. g., 15 seconds. However, notifications may also be delivered as the payload of any other request answered by the gamification server. This method of notifying the client makes the entire badge system asynchronous and decouples notifications from the events that have originally caused them.

⑤ **Notification display**. In the final step, the fact that a new badge—in this case B1—has just been unlocked is communicated back to the user. To that extent, a notification as shown in Figure 7.19 is displayed for a predefined number of seconds in the status panel of the user interface. Due to the asynchronous nature of the badge system, the amount of time that can elapse between the final action through which the user has unlocked a badge and the appearance of the notification can be between 10 and 30 seconds.

It should be noted that the entire business logic of unlocking badges is executed on the server-side of the HBM, thereby making it more difficult (albeit not impossible) for a potential attacker to claim ownership of a badge that he has not actually earned.

Presentation
An integral part of incorporating badges into a software lies in presenting earned badges as well as available badges to users in a fashion that facilitates their function as artifacts affording, e. g., goal setting, instruction and reputation [AC11]. To that extent, the current implementation of the gamification module provides two badge-centric displays. Firstly, the badges earned by a user are visualized on his profile page as illustrated in Figure 7.20. Depending on their level, badges may either by enclosed by a black border (Level 1) or a bronze, silver, or gold border (Levels 2-4). Furthermore, badges are grouped into different categories depending on the application area they most appropriately belong to. By hovering over a badge, its description is revealed. A circled arrow ⇧ affixed to the description indicates that a badge has a *successor*, i. e., a higher-level badge with similar completion logic but typically larger multipliers. Secondly, an additional display reachable via the “Show all” button in Figure 7.20 leads users to a list of all badges that the implementation currently provides. As shown in Figure 7.21, those badges that the user already possesses are depicted using their icon, whereas all other badges not yet unlocked are merely shown as a gray circle. There are many different ways how the presentation of badges could be extended, for instance by providing incremental feedback to users while they are working towards unlocking a badge, or incorporating secret or hidden badges that are not visible to the user at all or whose description is hidden until they have been unlocked.

7.4.5 Leaderboard

The leaderboard shown in Figure 7.22 provides an ordered list of all users on the current repository ranked by the number of points that they have earned. Each line of the leaderboard consists of the current user’s rank, his account name or full name, if provided, the number of points, and the profile picture or a placeholder if none has been uploaded. Clicking on a line opens the profile of the respective user in a new window and is presently the only way to access the profiles of others. The top-three users are highlighted through a larger font size and the use of special bronze, silver, and golden indicators. Finally, users may quickly identify their own position in the ranking by searching for the single line emphasized through its blue background color. A button in the

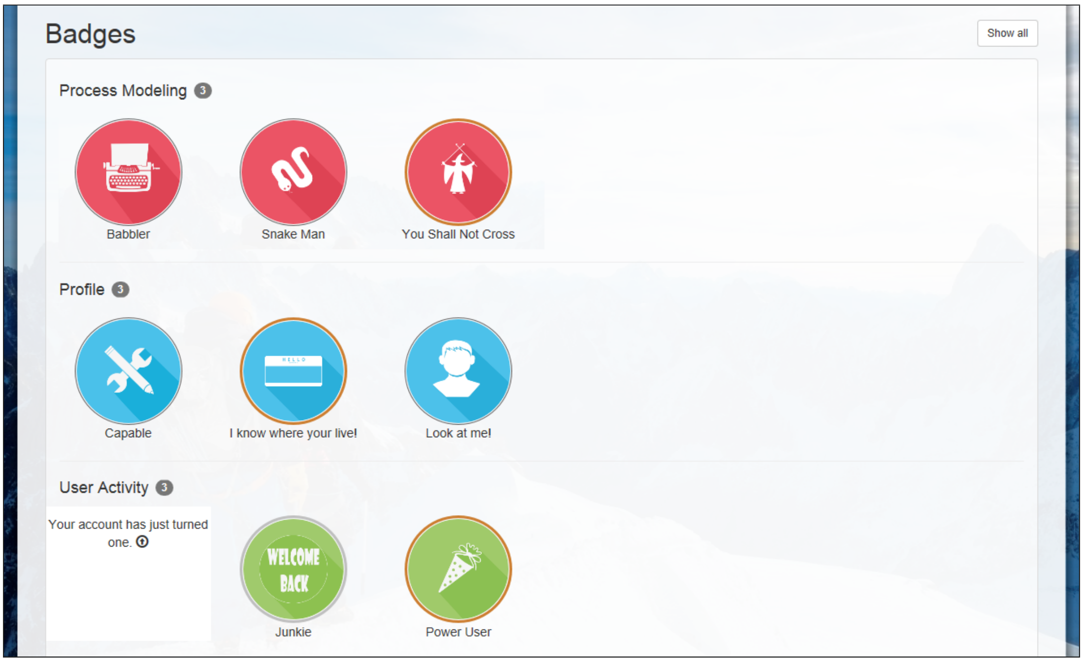


Figure 7.20: Implementation: Earned badges displayed in the user profile.

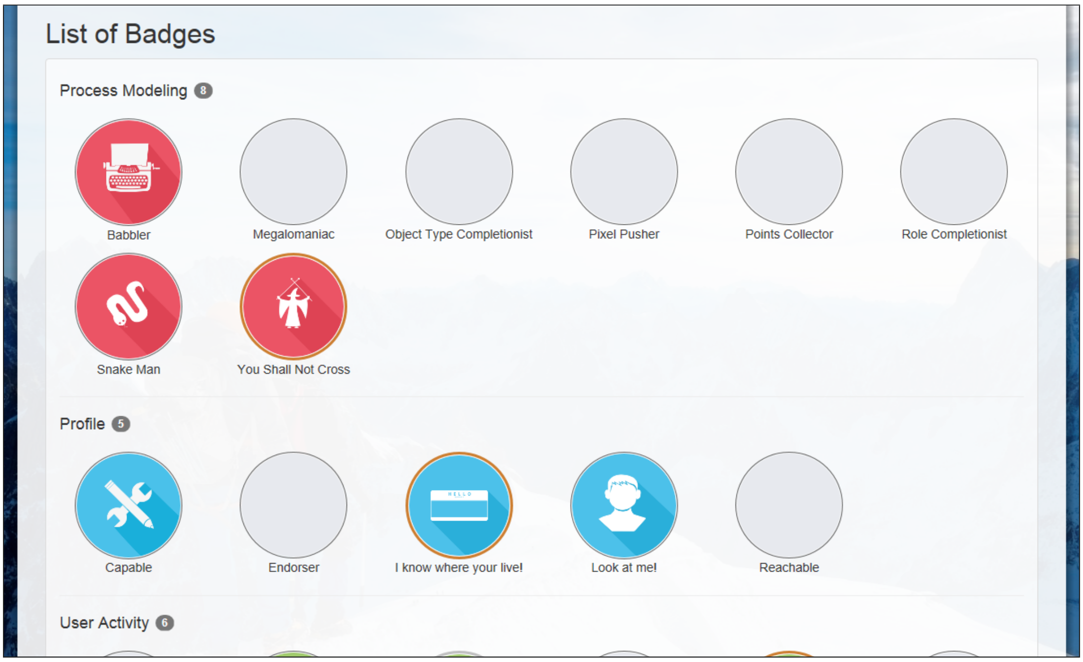


Figure 7.21: Implementation: Available badges displayed in the list of badges.

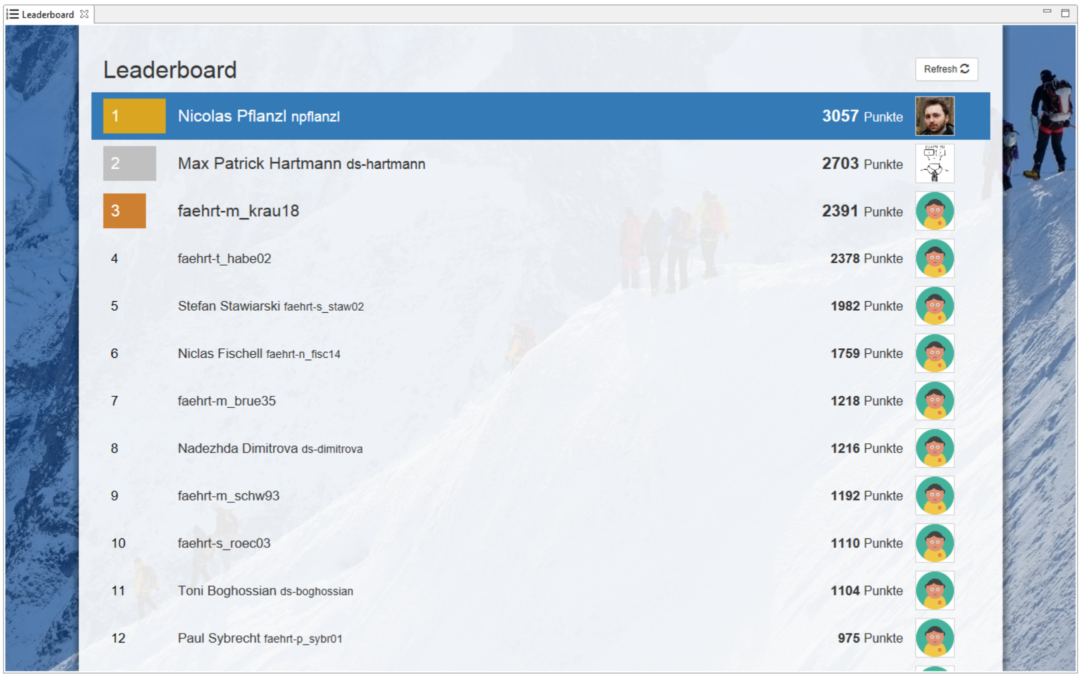


Figure 7.22: Implementation: Leaderboard with ranking by points.

upper right corner allows refreshing the leaderboard to reflect any changes that may have occurred since it was loaded. The current implementation is quite basic in its functionality and could be extended in many possible ways. For instance, the introduction of a weekly leaderboard ranking users only by the points earned in the last seven days could more effectively motivate new or relatively inactive users who perceive reaching a high position in the global leaderboard as too difficult. Furthermore, the ranking could update itself automatically at a given interval, with any changes in ranks or points being emphasized visually, e. g., through indicators such as ▼-3 and ▲2.

7.4.6 Introduction Page

Some of the implemented features serve the specific purpose of introducing new users into the functionality provided by the gamification module and informing existing users that this new functionality exists. To that extent, two particular events occur when users log into a server repository providing gamification for the first time. Firstly, the “New User” badge is immediately unlocked and users receive an according notification as previously shown in

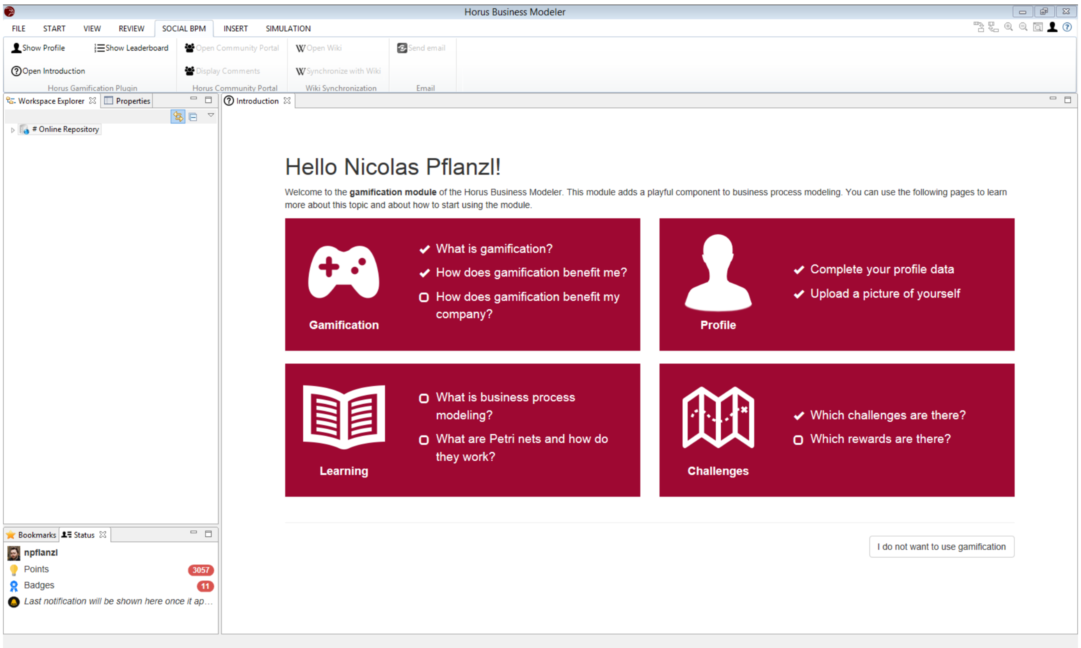


Figure 7.23: Implementation: Introduction overview page.

Figure 7.19. Clearly, this does not actually represent a notable achievement of the user and is thus only intended as a signal that gamification is activated. Secondly, the introduction page illustrated in Figure 7.23 is automatically shown to provide modelers with a means of receiving basic information about the functionality the gamification module provides and to give them an initial idea of the activities they can perform. For most entries on the introduction page, clicking on them reveals additional details as shown in Figure 7.24 and places a check mark next to the respective item. Besides the outlined approach, other methods for introducing modelers to the gamification module can be envisioned. For instance, rather than presenting information to users in a predominantly textual form (which conflicts with the game design principles discussed in Section 3.4.2), tutorials as conceptualized in Section 6.1 could serve as an alternative starting point. Another possibility would be to provide users with contextual help whenever they access a particular view or interact with a particular feature of the HBM for the first time.

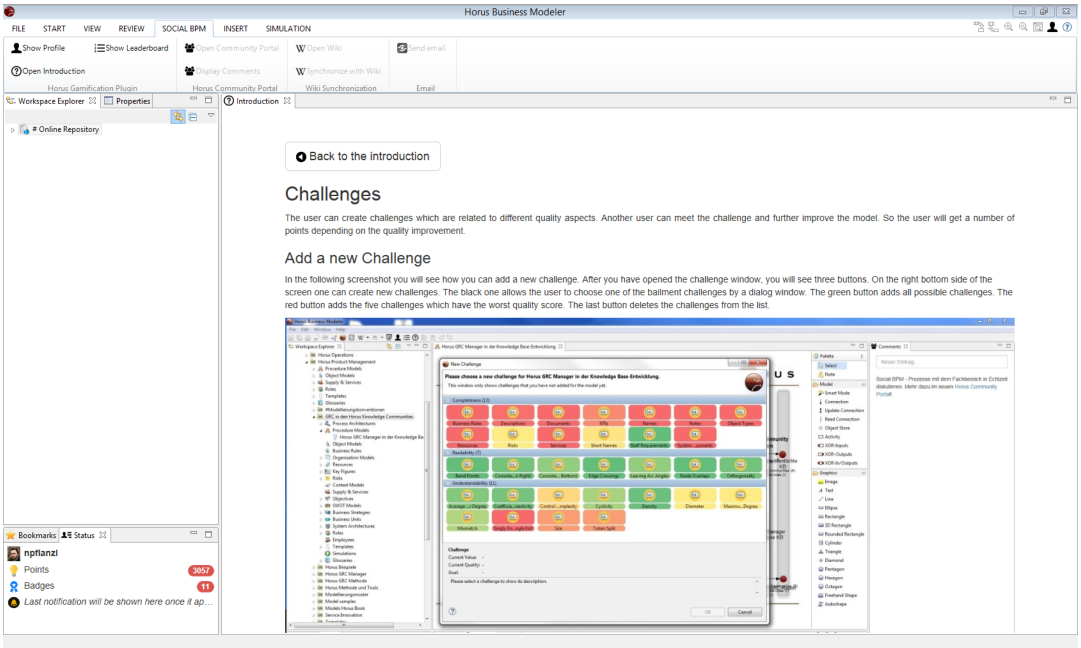


Figure 7.24: Implementation: Introduction details page for challenges.

8 Evaluation

In this chapter, the impacts of the implemented gamification module are evaluated against the goals of the research project defined in Section 4.3. To that extent, the chapter is structured as shown in Figure 8.1. As the evaluation is carried out by means of a quantitative approach, Section 8.1 first provides a brief overview of the statistical methods and tools that are utilized in the remainder of the chapter. Then, the results of two separate studies are presented, namely a *field experiment* in Section 8.2 and a *laboratory experiment* in Section 8.3. These two types of experiments differ as follows:

- Field research is conducted in real-world settings that closely match everyday situations, but allow little control over surrounding conditions [Rec13]. A field experiment is a special type of field research in which the impact of an independent variable (also called the *treatment* or *condition* [Goo10]) on a set of dependent variables is examined [Goo10]. Here, the field experiment was carried out as part of the lecture *Introduction to IS*, and its participants were thus students attending the lecture.
- Laboratory research is conducted in artificial settings that are distant from everyday situations, but allow for a high degree of control over surrounding conditions [Rec13]. A laboratory experiment is a special type of laboratory research in which the impact of an independent variable on a set of dependent variables is examined [Goo10]. Here, the laboratory experiment was conducted by means of a special process modeling task, and its participants were recruited from the students of the aforementioned lecture.

In both cases, the treatment was gamification, which was either be present or absent in the HBM, and the dependent variables were constructs related to model quality, system use, user experience, technology acceptance, and

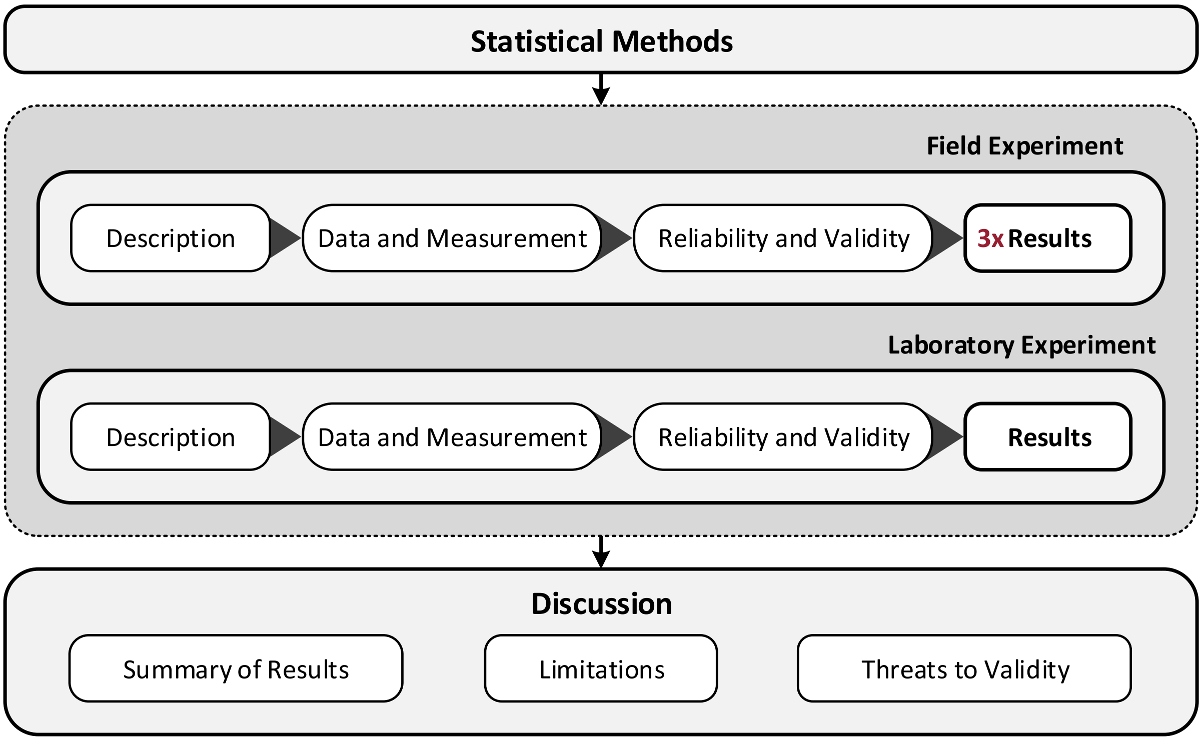


Figure 8.1: Structure of Chapter 8.

motivation. The structures of Section 8.3 and Section 8.3 are identical, and are comprised of the following contents. First, the settings of the experiments and the sets of participants are outlined. Second, the acquired datasets as well as the measurement instruments that were used to collect the data are described. In both experiments, data was collected by means of a survey, which was administered digitally in the field experiment, and through a paper-based questionnaire in the laboratory experiment. Additional datasets that were only collected in the field experiment include the work results of the participants, objective quality measurements of the resulting process models, and event histories extracted from the Horus server on which students worked. Third, the reliability and validity of the measurement instruments are examined to demonstrated the credibility of the collected data. Finally, the impacts of gamification are assessed by means of statistical hypothesis testing, and the findings are presented. For the field experiment, the discussion of results is divided into three separate sections (8.2.4-8.2.6) to improve readability.

The chapter then ends with a discussion of the findings in Section 8.4. Specifically, the results of both experiments are summarized and aligned with the challenges of Social BPM and the resulting goals that were defined for the created IT artifact in Chapter 4. Afterwards, the most important limitations of the experiments are outlined, and possible threats to their validity described along with the countermeasures that were implemented and possible avenues for further research.

8.1 Statistical Methods

As previously mentioned, experimental research is characterized by the existence of an independent variable—in this case gamification—that is hypothesized to influence a set of dependent variables. In both experiments that were conducted, the independent variable divides the collected data into two subsamples: the *experimental group* having worked with the implementation described in Chapter 7, and the *control group* having worked with the "conventional" HBM. The exact mechanisms through which participants were uniquely assigned to one of the two, independent groups are described later in this chapter. To examine whether the independent variable causes any differences between both groups that are not just the result of pure chance, tools and methods for the testing statistical hypotheses are used [Leh97]. In *hypothesis testing* two assertions are formulated about the data at hand: the *null hypothesis* stating that the independent variable has no impact on the dependent variable, and its complementary *alternative hypothesis* claiming the opposite to be the case [Pag13].

Many testing procedures that are employed in practice are *parametric*, meaning that they are based on certain assumptions about the distribution, sample size, and other aspects of the underlying data. For instance, one requirement of the widely-used *independent t-test* is that the data of the two groups being compared follows a normal distribution [Fie09], which can be assessed using the Kolmogorov-Smirnov [Kol33] and Shapiro-Wilk [SW65] tests. If this assumption is violated, *nonparametric* testing procedures with fewer requirements can be used as an alternative. The non-parametric equivalent to the independent t-test is the *Mann-Whitney-U test*, which is based on prior work by Wilcoxon [Wil45] and was first proposed in 1947 by Mann and Whitney [MW47]. The

Table 8.1: Evaluation: Report schema for statistical hypothesis tests.

Column	Description
Variable	Name of the variable for which the test is performed.
Homogeneity	✔: homogeneous variances. ✖: heterogeneous variances.
U-Value	Test statistic of the Mann-Whitney-U test.
Z-Value	Standardized representation z of the test statistic.
p-Value	p-value determined by the Mann-Whitney-U test.
Significance	*: p-value < 0.1. **: p-value < 0.05. ***: p-value < 0.01.
Effect	Effect size r computed from z and the sample size.
Size	X: $r < 0.1$. S: $r \geq 0.1$. M: $r \geq 0.2$. L: $r \geq 0.3$

null hypothesis of this test states that the probability for one observation of the experimental group to exceed one observation of the control group is as high as vice versa. Accepting this hypothesis would for instance mean that when picking one random process model from the experimental group and one from the control group, no statement could be made about which model can be expected to exhibit a higher quality. Consequently, the alternative hypothesis claims that one group is stochastically greater than the other with regard to the dependent variable, meaning that its cumulative distribution function yields smaller probabilities for any given value. In informal terms, this means that if the test results allow for a rejection of the null hypothesis, values that are randomly drawn from one subsample can be expected to exceed values randomly drawn from the second subsample. In this context, this could for instance indicate that gamification allows modelers to create process models of significantly higher quality.

As it will be shown in the following sections, the collected datasets violate the assumption of being normally distributed for most measured variables, and thus the Mann-Whitney-U test is applied. To that extent, results of this test are reported with the structure shown in Table 8.1. Here, the columns *U-Value* and *Z-Value* are test statistics computed as part of the test, and the remaining columns are used to report on the following findings:

- **Homogeneity of variances** indicates whether the variances of a random variable are equal across the experimental group and the control group.

This characteristic is also called *homoscedasticity*, and its counterpart, i. e., inequality of variances, is referred to as *heteroscedasticity* [Kli11]. Possible reasons for the latter include random error, outliers, and non-normality in the stochastic variable for one of the groups. While the Mann-Whitney-U test does not require *homoscedasticity* to be effective, it has been shown that a violation of this property together with non-normally distributed data may lead to an increased risk of rejecting the null hypothesis despite its correctness [Zim04]. Thus, any findings that are presented in conjunction with unequal variances have to be interpreted with care. Homoscedasticity can be examined using Levene's test [Lev60], and the results of this test are indicated as ✔for homogeneous variances at the significance level 0.05, and ✖ otherwise.

- **Statistical significance**. Following an informal definition of the American Statistical Association, a *p-value* represents "the probability under a specified statistical model that a statistical summary of the data [. . .] would be equal to or more extreme than its observed value" [WL16, p. 131]. In the context of statistical testing, low p-values may indicate that a proposed statistical model for some dataset—for instance a null hypothesis stating the absence of an effect—is incompatible with the actual data, and therefore may be rejected. Furthermore, a finding is declared to be statistically significant if the p-value obtained from a test is below a certain threshold, the so-called *significance level* or *alpha level* α [Joh13]. While the threshold most commonly used as a basis for making scientific claims is 0.05 [WL16], other values such as 0.1, 0.01, 0.005, and 0.001 [Coh92, Joh13] can be found as well. In general terms, the significance level indicates the likelihood of rejecting the null hypothesis despite its truthfulness. In the tests performed in this section, three significance levels are used, namely 0.1, 0.05, and 0.01, with the first and the last serving as less and more rigorous alternative to the most common value. Significance at these three levels is indicated as *, **, and ***respectively.

- **Effect size**. Despite the importance of statistical significance for hypothesis testing, p-values do not allow researchers to make any statements about the sizes of the observed effects, and can thus not guarantee the actual relevance of results [WL16]. Thus, the American Psychological Association requires researchers to provide measures of *effect size* in most research [FMR12]. Academic literature provides many different conceptualizations of effect size estimates that are appropriate for different tests and under varying conditions [Coh92, FMR12]. One example that is used for the Mann-Whitney-U test is based on the standardized test statistic z and can be computed as Pearson's correlation coefficient $r = |z/\sqrt{N}|$ [FMR12], with r denoting the effect size, and N the size of the entire sample including experimental and control group. Originally, Cohen suggested that $r > 0.1$, $r > 0.3$, and $r > 0.5$ correspond to small, medium, and large effect sizes, respectively. However, a recent meta-analysis found these thresholds to be inappropriate given the results of 87 studies, with only 3% satisfying the conditions for a large effect [GS16]. Thus, the authors suggest more realistic guidelines using 0.1, 0.2, and 0.3 as corresponding thresholds. These are also employed in this evaluation, and small, medium, and large effect sizes are indicated as S, M, and L, respectively. Lastly, X is used to signify a statistically significant result with an effect size smaller than the minimum threshold.

Unless noted otherwise, all analyses reported in the following have been conducted with the the statistics software IBM SPSS[1] following the instructions provided in relevant textbooks [Fie09, Pag13, CF14b, GS14]. Most notably, this includes tests for normality, homogeneity of variances, hypothesis testing, and measurement reliability. Supplemental analyses have been performed using the software tools SmartPLS[2] Version 3.2.4 and IBM SPSS Amos[3] Version 24.0 and will be highlighted explicitly as such where applicable.

8.2 Field Experiment

In this section, the results of a field experiment carried out in the context of this research project are reported. To that extent, Section 8.2.1 first describes the experiment itself, including its context and participants. Section 8.2.2 then

1 See: https://www.ibm.com/analytics/us/en/technology/spss/ Version 23. Last accessed: 2017-04-19.
2 See: https://www.smartpls.com. Last accessed: 2017-04-19.
3 See: https://www.ibm.com/de-de/marketplace/structural-equation-modeling-sem. Last accessed: 2017-04-24.

outlines the different datasets that were collected during the experiment, and the reliability and validity of these datasets is examined in Section 8.2.3. The section ends with the analysis of the obtained data and the presentation of the results in Section 8.2.4, Section 8.2.5, and Section 8.2.6.

8.2.1 Description

The field experiment was conducted as part of the lecture *Introduction to Information Systems* at the Department for Information Systems of the University of Münster, which is part of the Information Systems Bachelor curriculum and is regularly attended in the first semester. The main purpose of this course is to introduce students to the most important aspects of IS as a domain, and to provide them with a realistic picture of what their future professional careers in this discipline may look like. Thus, students should be enabled to critically reflect and reaffirm their choice of study course as early as possible, or to transition to another program.

Historically, *Introduction to IS* was taught in a "traditional" lecture format closely adhering to a standard IS textbook, focusing on terminology, nomenclature, and definitions, and reviewing whether students have reached the predefined learning goals through a written exam based on knowledge reproduction. As this senior academic staff concluded that this format was unable to achieve the aforementioned lecture goals, the lecture was reorganized in 2013 around the following principles [PBSV15a]:

1. **Experience-oriented case study**. By means of a case study, students should be able to experience their possible future jobs in the IS domain.

2. **Independent group work**. Working together on the case study in groups should allow students to develop soft skills and learn about and overcome some of the most common challenges of teamwork.

3. **Complement case study with traditional lectures**. Through complementary lectures given by the professors of the Department for Information Systems, students should be able to acquire foundational knowledge about the IS domain.

4. **Close link between lecture and case study**. The knowledge acquired in lectures should serve as a valuable input for the case study. Consequently, each lecture should provide concrete tasks for continuing work on the case study.

5. **Well-aligned excursion**. The case study should be based on a real organization, and an excursion to this enterprise at the beginning of the lecture should give students a palpable introduction into the upcoming case study work as well as the lecture contents.

6. **Alternative approaches to grading**. Rather than through a traditional exam, students should be graded according to the results of their case study work and complementary reports and presentations.

As this list shows, the case study plays a central role for the reorganized lecture. Due to the emphasis the IS curriculum at the University of Münster and the research activities of the Department of IS put on Business Process Management, this part of the lecture was conceptualized as a process modeling case study designed around the car manufacturer Ford situated in Cologne. Specifically, students were subdivided into groups with a standard group size of five students, each of which was assigned to a particular department of the firm, such as production, logistics, marketing, or accounting. Over the course of the semester, these groups had to create multi-perspective process descriptions—modeling not only procedures, but also business objects, roles, and organizational units—of the most important tasks of their departments. Inputs for that purpose were provided through an excursion to the aforementioned car manufacturer, nine lectures on different aspects of the IS discipline held by the professors of the department, and independent internet research. As a full description of the case study is outside the scope of this thesis, the reader is referred to [PBSV15a] and the case study task description in Section C.1 of the appendix for further details.

As part of the ongoing cooperation between the DBIS Group and the Horus software GmbH (see Section 5.1), students worked with the Horus Business Modeler (HBM) and the required infrastructure was provided and maintained by Horus. After two semesters in which the case study was conducted as described, some changes were implemented based on feedback received in the lecture evaluation. Most notably, instead of letting students work in groups, they were required to work individually. This change was made because many groups suffered from absenteeism, i. e., members who contributed very little or nothing at all. Furthermore, instead of randomly assigning students to particular departments of the car manufacturer, they were allowed to decide autonomously on which processes they would like to focus. The reason for this alteration was that students of previous semesters found the business processes of some departments much more difficult to describe than those of others. One other important change resulted from the fact that students always worked with the most up-to-date version of the HBM that was available in each semester. Consequently, with the availability of an initial version of the Horus gamification module in Version 2.5.4, students also received the opportunity to work with the implemented gamification concept described in Chapter 7. However, this was not mandatory, and all participants could freely choose between working online with the support of gamification, or offline in their local file system without, although the consequences of this choice were not communicated to students.

A final difference between the four iterations of the case study lies in the requirements for work results that were communicated to the the students. Whereas in the first two years no requirements were specified at all—and thus groups did not work towards concrete goals—this was changed in the latter two iterations to equalize the efforts invested by all students. Specifically, in the winter term 2015/16, hard goals for the minimum number of process models (2), object models (1), and organizational diagrams (1) were specified. Students did not receive additional information about the expected minimum quality of their submissions and were not instructed to use the available quality feedback. Hard goals were also provided in 2016/17, although the minimum number of process models was raised from 2 to 4. Furthermore, the course participants of this semester were given additional hints and recommendations (i. e., soft goals) for when models can be considered “good enough” to be submitted for a passing grade. This supplemental information addressed the expected minimum model size, the need to write description texts for model elements and create references between process models, organizational diagrams, object models, and roles, and naming guidelines for model elements. Furthermore, students received explicit instructions to use the provided model quality feed-

Table 8.2: Field experiment: Sets of participants.

Semester	Students	Work	Dept.	HBM Ver.	Gamif.	Requirements
2013/14	162	Group	fixed	2.5.0	✖	none
2014/15	200	Group	fixed	2.5.2	✖	none
2015/16	183	Individual	choice	2.5.4	✔	hard
2016/17	250	Individual	choice	2.6.1	✔	hard+soft

back and the available help features. More detailed information and the exact wording of these soft goals can be found in Section C.2 of the appendix.

A summary of the four instantiations of the *Introduction to IS* lecture which were based on the case study described in [PBSV15a] and for which data was collected is provided in Table 8.2 together with the number of students who attended the course in each semester. Due to the conceptual changes the lecture has undergone over time, the results that will be presented in the following may not be perfectly comparable across years, which is why the evaluation will focus on inter-group differences that can be determined in a single year.

8.2.2 Data and Measurement

In this section, the three datasets that were collected during the field experiment are described. These datasets relate to the quality of the process models that participants created, their actual use of the HBM, and their user experience while working with the software.

Model Quality

The quality of the models that were created by the students was assessed by computing the complete set of 32 quality metrics (see Section 6.1.1) for all submitted business process models. To that extent, a *quality report* convenience feature was integrated into the HBM; this function creates a complete report about the quality of all models (process models and otherwise) that are included in a particular workspace. Before conducting the data analysis, two preprocessing steps were carried out.

Firstly, all models with a size below a certain threshold were excluded, as they are neither likely to exhibit quality defects, nor to include any meaningful content. Thus, an inclusion of such models would lead to an overestimation of the “real” average model quality. As mentioned in Section 2.7.3, [KM96] recommends a minimum model size of 5 elements, and thus all diagrams with 4 elements or less were excluded. For the winter term 2016/17, additional models were removed from the analysis; specifically, students were provided with a top-level process architecture, and were instructed to create all of their own models as refinements of the former. Therefore, this architecture represents a constant occurring in the submissions of all students, and it was thus exempted from further consideration.

Secondly, for the last two iterations of the case study, it was necessary to distinguish those models that were created with the support of gamification, and those that were created without. This was trivial for the winter term 2015/16, as the type of submission allows for a reliable distinction between these two groups. However, in the winter term 2016/17, all students were required to eventually import their models into the gamification-enabled gamification server, even if they had previously modeled on their local machines. Thus, many course participants performed the largest part of the case study work offline, proceeded to uploaded their models, and then made some final adjustments based on the feedback of the gamification module. As a consequence, a binary distinction between individuals who have used gamification *at all times* and those who did *not at all* is not possible. Instead, students were allocated to one of the two groups based on the *intensity* of their gamification utilization. As this construct is not measured explicitly, the number of gamification points was used as a surrogate, since earning points requires the actual use of the gamification module. In particular, the following two criteria were employed:

- **Earned points**. To fulfill the hard requirements of the case study, students had to create at least 4 process models. As 17 of the 32 available quality metrics were enabled on the repository and 10 points could be earned per metric, $17 * 10 * 4 = 680$ points could be earned under the assumption of perfect model quality. As the average model quality of all non-trivial models was determined to be 57%, realistically, students should have been capable to earn at least 50% of the possible score, i. e., 340 points. On this basis, a student was classified as a gamification-user if he earned not less than half of the reasonable minimum score, i. e., 170 points or more.

Table 8.3: Field experiment: Model quality dataset overview.

Semester	Models *Total*	*Excluded*	*Included*	Gamification	Included
2013/14	452	27	425	✖	425
2014/15	480	35	435	✖	435
2015/16	559	9	550	✔	277
				✖	273
2016/17	1128	225	903	✔	560
				✖	343

- **Model points**. For each user, the total number of points the user should at least have earned if all work was performed on the gamification-enabled repository was calculated from the set of submitted models. On this basis, a student was classified as a gamification-user if his actual number of points was not less than a third of the score he should theoretically possess. Conversely, it can reasonably be assumed that individuals with a lower ratio than that conducted most of the case study work without gamification, and thus without earning points as a reward.

These two criteria were used in a conjunctive fashion, meaning that both had to be fulfilled to be classified as a gamification user. Furthermore, it should be noted that they are subjective in nature, and thus another method for distinguishing between gamification and non-gamification users may lead to different results than those presented in Section 8.2.4. A summary of the model quality datasets obtained from the four instantiations of the *Introduction to IS* lecture is given in Table 8.3.

System Use

Data about the students' use of the HBM was collected directly from the Oracle RDBMS used by the Horus server repository the students were working on. This data is made available by the event tracking facilities that were implemented as part of the gamification module (see Section 7.2.1) and is stored in the HORUS_EVENTS database table (see Table 7.1). By examining this data, it is possible to make statements about when, how often, how long, and with which behavioral patterns the students interacted with the HBM. For

Table 8.4: Field experiment: Sample system use records.

ID	Name	Event Type	Date and Time
1301	t_acke04	user.login	2015-11-25, 10:34:19
1301	t_acke04	reward.badge.unlocked	2015-11-25, 10:34:20
1301	t_acke04	projectManager.modelCreatedAsCopy	2015-11-25, 10:35:20
1301	t_acke04	editor.modelOpened	2015-11-25, 10:36:04
1301	t_acke04	editor.petriNet.propertyChanged.lineWidth	2015-11-25, 10:36:16
1301	t_acke04	editor.petriNet.propertyChanged.Node.Name	2015-11-25, 10:36:16
1301	t_acke04	editor.petriNet.propertyChanged.capacity	2015-11-25, 10:36:16

instance, from the excerpt of a sample modeling session shown in Table 8.4, it can be concluded that the user *t_acke* with the user ID *1301* has logged into the Horus server on the 25th November 2015, upon which a badge—most likely the “New User” badge—was unlocked. The user has then proceeded to copy a model, which he then opened to change certain properties—line width, name, and capacity—of some model elements.

Such event data was collected in the two semesters where gamification was available, i. e., the winter terms 2015/16 and 2016/17. To examine the impacts of gamification, only such data which allows for a clear distinction between users having worked with gamification and those having modeled without is of any value. This is possible for the winter term 2015/16, as the data allows identifying a subset of students having worked on the server repository (i. e., online), albeit without the support of game design elements. However, no event data is available for users who have worked locally (i. e., offline). Such a distinction cannot be made for the winter term 2016/17, as students were required to submit their work results on the gamification-enabled server repository. Thus, while many course participants worked online throughout the entire case study, others modeled locally, uploaded their models to the server briefly before submissions, and then complemented and adjusted their models based on the gamified quality feedback. Thus, for many users, event data was only recorded for these brief periods in time, and full records of events for students having worked without gamification are not available. Therefore, the winter term 2016/17 was not considered for further analysis regarding system use, as the validity of any results gained from this data would have been highly questionable. A summary of the usable data that forms the

Table 8.5: Field experiment: System use dataset overview.

Semester	Students *Total*	*Excluded*	*Included*	Gamification	Included
2015/16	136	14	122	✔	79
				✖	43

basis of the results presented in Section 8.2.5 is provided in Table 8.5. From the total number of 136 students having generated 191.379 event records, the data of 14 users was excluded, as they did not submit any work results, or their overall activity indicated that they predominantly worked offline. As a threshold for the second condition, students were required to have caused the creation of at least 200 event records to be included in the analysis.

As the original event data as illustrated in Table 8.4 does not contain any quantitative indicators suited for analysis, the number of event instances per event type and user was first counted as shown in Table 8.6. Furthermore, as the identifiers of events allow arranging event types in a hierarchical fashion according to their shared prefixes, event counts were also summed up in a hierarchical fashion. For instance, the number of events counted for the event type events.editor is determined as the sum of all occurrences of events with a type matching the regular expression events.editor.*. Lastly, the following additional measures were computed from the original event data to enable further tests going beyond simple event count values:

- *Distinct event types*: Number of distinct types of events fired by a particular user. This measure represents the degree to which a student has used the breadth of the functionality offered by the HBM.
- *Sum of session lengths in seconds*: For each modeling session, starting with a login event and ending with the last event recorded before the next login, the total number of seconds the session has lasted is determined and summed up across all sessions. This measure represents the total aggregated time a student has used the HBM.
- *Average session length in seconds including empty sessions*: Identical to the previous measure, but determines the average session length rather

Table 8.6: Field experiment: Sample system use records (aggregated).

ID	Name	Event Type	Event Count
1301	t_acke04	events	1190
1301	t_acke04	events.editor	1113
1301	t_acke04	events.editor.modelClosed	45
1301	t_acke04	events.editor.modelOpened	45
1301	t_acke04	events.editor.modelSaved	36
1301	t_acke04	events.editor.petriNet	987
1301	t_acke04	events.editor.petriNet.created	221

than the sum. Includes sessions in which students have logged in, but did not subsequently perform any additional actions.

- *Average session length in seconds excluding empty sessions*: Identical to the previous measure, but excludes "empty" sessions.
- *Average decision time*: Average time in seconds between two subsequent events of the same user. This measure represents how quickly the user was capable of deciding which action to perform next.

User Experience

Besides investigating whether the integration of gamification into a process modeling tool allows its users to create higher-quality models and increases their engagement with and actual use of the software, an additional goal of the evaluation was to determine whether the perception of the Horus Business Modeler itself by students is altered in the presence of game design elements. Since the reorganization of the *Introduction to IS* lecture, an online evaluation by means of a 55-item survey was conducted at the end of each semester using the software *LimeSurvey*[4]. Its purpose was to examine whether students were satisfied with the concept and instantiation of the course, or whether there was need to implement any changes. Two example modifications that resulted from this evaluation were the switch from group to individual work, and the inclusion of a factory tour in the excursion. This survey also included question

[4] See: https://www.limesurvey.org/. Last accessed: 2017-04-29.

items related to the *Components of User Experience (CUE)* model, an approach aiming to explain why some individuals prefer certain products or systems over others [TM07b]. In doing so, the CUE go beyond mere functionality and usability by considering the following aspects of user experience:

- **Perceptions of instrumental qualities**: These factors relate to how well a system allows its users to accomplish the intended tasks and are concerned with *usability* (effectiveness) and *usefulness* (efficiency).
- **Perceptions of non-instrumental qualities**: These factors relate to the look and feel of a system and are concerned with *visual aesthetics*, *haptic quality*, and *status and identification*.
- **Emotional user reactions**: Depending on the instrumental and non-instrumental qualities of a system, its use may be accompanied by certain emotional responses of its users. For instance, whereas a responsive system with innovative design may lead to positive emotions, an unresponsive, visually unpleasing system may cause the opposite. Consequently, these factors are related to *positive emotions* and *negative emotions*.
- **Consequences of the user experience**: Together, instrumental and non-instrumental qualities as well as emotional user reactions shape the overall appraisal of the system by users and their decisions about its future use. Factors related to this include *overall system judgment*, *behavioral intention* (continued use of the system), and *loyalty* (intention to use alternative systems).

The CUE model was operationalized by means of the meCUE questionnaire[5], whose structure, reliability, and validity have previously been demonstrated [TM07b]. The questions were used in the randomized order provided by the creators of the questionnaire and reformulated to relate to the HBM and the case study. Furthermore, some question items were dropped to reduce the size of the overall survey, or as they were perceived to be irrelevant or inappropriate for the given context. For instance, other than a new and expensive high-end smartphone, the HBM is unlikely to act as a status symbol. A summary of the

[5] Available from: http://mecue.de. Last accessed: 2017-04-16.

Table 8.7: Field experiment: User experience survey overview.

Construct	Items *Available*	*Utilized*	*Analyzed*	Codes	Scale
Usability	4	4	3	U1-U3	7-point Likert
Usefulness	3	3	3	N1-N3	7-point Likert
Aesthetics	3	1	1	A1	7-point Likert
Status	3	-	-	-	7-point Likert
Commitment	3	-	-	-	7-point Likert
Positive Emotions	6	4	4	EP1-EP4	7-point Likert
Negative Emotions	6	5	4	EN1-EN4	7-point Likert
Usage Intention	3	3	2	NI1-NI2	7-point Likert
Loyalty	3	2	2	L1-L2	7-point Likert
Overall Judgment	1	1	1	G1	School grade

number of items remaining after deletion is provided in Table 8.7, with the last column indicating the question codes of the respective items. Note that the fourth column in this table indicates how many items were ultimately used for the analysis described in Section 8.2.6 after the checks for reliability and validity documented in Section 8.2.3. Most questions were measured on a 7-point Likert scale ranging from "complete rejection" (1) to "complete agreement" (7) to statements such as "The design of Horus is attractive", "Horus is easy to use", or "Horus frustrates me". An exception from this is the overall judgment of the HBM, which was measured as a school grade with the following scale (ranging from best to worst grade): $1.0, 1.3, 1.7, 2.0, 2.3, \ldots, 3.7, 4.0, 5.0$. For additional information on the survey, including the full wording of all questions, the reader is referred to Section C.4 in the appendix.

While the survey was also conducted in the winter terms 2013/14 and 2014/15, the respective data has not been included in the analysis as the HBM has undergone numerous changes in these periods that include, but also go significantly beyond gamification. Therefore, comparing these historic results to those of 2015/16 has little explanatory power due to the difficulty of isolating the effects of gamification from those of other changes and bug-fixes. The evaluation in the winter term 2016/17 did not include the meCUE items, and thus no data is available for the most recent instantiation of the lecture. Since participation in the evaluation is voluntary, only a fraction of the students having attended the course filled out the survey as summarized in Table 8.8.

Table 8.8: Field experiment: User experience dataset overview.

Semester	Responses *Total*	*Excluded*	*Included*	Gamification	Included
2015/16	83	4	79	✔	51
				✖	28

Four datasets were excluded, as the respective students either did not provide answers for most (or all) of the questions, or responded in a dubious manner, e. g., by only using extremal responses or using the same response for all questions. Lastly, due to the possibility of skipping individual questions without providing an answer, the number of responses ranges from a minimum of 49 to a maximum of 51 for the experimental group with gamification, and from a minimum of 26 to a maximum of 28 for the control group without gamification. No attempt was made to restore the missing data; instead, subsequent analyses are limited to the complete datasets for each respective construct.

8.2.3 Reliability and Validity

When conducting quantitative analyses of collected data, researchers must be able to show that their measurements are accurate. Otherwise, any conclusions that are drawn from the data—no matter how carefully the selected statistical methods are selected and applied—may be questioned [Rec13]. To overcome this problem, two key properties of the utilized measurement instrument (i. e., survey) must be shown, namely its *reliability* and *validity*. Whereas reliability is "concerned wit the ability of an instrument to measure consistently" [TD11, p. 53], validity describes "the extent to which an instrument measures what it is intended to measure" [TD11, p. 53]. Consequently, unreliability can cause random differences in multiple observations of the same measure, and invalidity means that measurements may not properly represent the real-world phenomena they are intended to quantify. For a measurement instrument to be valid, it must also be reliable; however, the inverse is not true and reliability can be given in the absence of validity [TD11]. In this section, the reliability and validity of the meCUE questionnaire that was employed to investigate the user experience of students working with the gamified and non-gamified versions of Horus are examined. For the other two datasets, both properties are assumed to be given due to the objectiveness of their associated measures.

Reliability

While there are different types of reliability (cf. [SBG04, Rec13]), the most relevant one for this evaluation is *internal consistency*, the "extent to which all the items in a test measure the same concept or construct" [TD11, p. 53]. Consequently, if the responses in a survey have a high internal consistency for the items measuring a particular construct, this means that the related questions are interrelated and do indeed measure the same concept. Inversely, a low consistency may indicate that one or more question items do not actually measure what they should [TD11]. Therefore, internal consistency should always be examined when appropriate before conducting data analysis.

The most popular and widely-used measure of internal consistency is the *coefficient alpha*, which was proposed by Cronbach in 1951 [Cro51] and is hence also referred to as "Cronbach's α [Pet94]. Conceptually, this coefficient measures the intraclass correlation of all items belonging to the same scale and can assume values between 0 and 1. Academic literature offers a variety of guidelines for the minimum threshold for acceptable reliability, with most proposals being set at 0.7 or higher [Pet94]. Further information can be gained from a comparison of the *inter-item correlations* of the responses for all questions belonging to the same construct. Specifically, it is recommended to consider dropping items with a high frequency of correlations below the threshold 0.3, as this indicates that something different is actually being measured [Fie09]. Results of the analysis of internal consistency for the user experience survey data are as follows:

- *Usability*: Cronbach's α was calculated as 0.816 with all four items included. No inter-item correlation was below the threshold of 0.3, and thus no item was deleted.

- *Usefulness*: Cronbach's α was calculated as 0.804 with all three items included. No inter-item correlation was below the threshold of 0.3, and thus no item was deleted.

- *Aesthetics*: As this construct is only measured by one item, no reliability analysis was conducted.

- *Positive emotions*: Cronbach's α was calculated as 0.762 with all four items included. One item exhibited varied correlations with the remaining items ranging from 0.191 to 0.671. However, as its removal does not result in a strong increase of Cronbach's α, the item was not deleted.

- *Negative emotions*: Cronbach's α alpha was calculated as 0.785 with all five items included. One item exhibited varied correlations with the remaining items ranging from 0.246 to 0.62. As its removal results in a strong increase of Cronbach's α to 0.83, the respective item was deleted.

- *Usage intention*: Cronbach's α was calculated as 0.744 with all three items included. No inter-item correlation was below the threshold of 0.3, and thus no item was deleted.

- *Loyalty*: Cronbach's α was calculated as 0.474 with all three items included. One item exhibited low correlations consistently below 0.1 with the remaining items. Removal of this item has resulted in an increase of Cronbach's α to 0.662.

As Cronbach's α for the construct *loyalty* was still below the threshold 0.7, an Exploratory Factor Analysis (EFA) [Fie09] was conducted for all six indicators for *usage intention* and *loyalty* due to their conceptual similarity. Results of this analysis demonstrate that all items with the exception of one indicator for *loyalty* properly load onto a single factor. In other words, this means that the respective items are highly correlated with each other and measure different dimensions of the same construct [Fie09]. This construct is henceforth referred to as *behavioral intention*, thus mirroring the terminology used in the TAM [VB08]. Cronbach's α for this new factor was calculated as 0.833 with all five items included. No inter-item correlation was below the threshold of 0.3, and thus no item was deleted. With these modifications in place, Cronbach's alpha is larger than 0.8 for all constructs except *positive emotions*, for which it still surpasses the minimum level of 0.7, and thus the utilized measurement instrument can be said to exhibit internal consistency.

Validity

As with reliability, there exists a variety of different types of validity (cf. [SBG04, Rec13]). For instance, *face validity* relates to the question whether measurement indicators are appropriate for the quantification of constructs they should represent, and *content validity* refers to the extent to which measurement items assess the theoretical concept they should represent in its entirety [Rec13]. Both types of validity can be assessed by experts and will be assumed to be present in the following due to the use of a previously-published, validated questionnaire. More importantly, *construct validity* is concerned with whether the "instrument items selected for a given construct are, considered together and compared to other latent constructs, a reasonable operationalization of the construct" [SBG04, p. 388] (also see [CM55]). It consists of two separate aspects, namely *convergent validity* and *discriminant validity*. Whereas the former is given if the items measuring the same construct converge (i. e., they are similar and correlate strongly), the latter requires items measuring one construct to be dissimilar from the items measuring other constructs [Rec13]. This is similar to reliability, but differs from the latter insofar as the examination is not only restricted to indicators of the same construct, but also considers items measuring other variables [SBG04].

One possible way to test both convergent as well as discriminant validity of a measurement instrument is to assess its *factorial validity* [SBG04]. To that extent, a factor analytic technique is applied, and the results examined with regards to how individual items load onto constructs. If factorial validity—and thus also convergent and discriminant validity—is given, measurement indicators should only properly load onto those variables for which this is expected without exhibiting strong cross-loadings onto other constructs. For the meCUE questionnaire employed in the field experiment, this type of validity was investigated by means of a Confirmatory Factor Analysis (CFA) [Jör69] conducted with SmartPLS, and a reflective measurement model underlying the analysis was specified following [TM07b]. The factor loadings delivered by the analysis are presented in Table 8.9. As it can be seen, two indicators—namely U4 and NI3—have factor loadings for their constructs of less than 0.6 that are also considerably smaller than the loadings of the remaining items. Thus, these indicators have been dropped from the analysis. After a repeated CFA, no factor loading is smaller than 0.664, and the data indicates only one problematic

Table 8.9: Field experiment: User experience survey factor loadings.

Item	Usability	Usefulness	Pos. Emotions	Neg. Emotions	Beh. Intention
U1	**0.825**	0.531	0.286	-0.633	0.445
U2	**0.830**	0.479	0.203	-0.550	0.377
U3	**0.933**	0.580	0.327	-0.706	0.550
U4	0.598	0.227	0.109	-0.518	0.318
N1	0.374	**0.802**	0.047	-0.347	0.382
N2	0.641	**0.889**	0.233	-0.543	0.703
N3	0.363	**0.861**	0.051	-0.346	0.343
EP1	0.388	0.267	**0.662**	-0.292	0.277
EP2	0.187	0.122	**0.788**	-0.135	0.372
EP3	0.045	-0.029	**0.780**	-0.104	0.082
EP4	0.224	0.064	**0.885**	-0.187	0.262
EN1	-0.711	-0.510	-0.306	**0.865**	-0.578
EN2	-0.478	-0.492	-0.161	**0.743**	-0.281
EN3	-0.603	-0.303	-0.191	**0.838**	-0.266
EN5	-0.637	-0.359	-0.111	**0.825**	-0.409
L1	0.241	0.368	0.153	-0.190	**0.670**
L2	0.582	0.584	0.264	-0.540	**0.880**
NI1	0.485	0.582	0.254	-0.419	**0.874**
NI2	0.482	0.548	0.373	-0.425	**0.904**
NI3	0.162	0.195	0.341	-0.271	0.553

cross loading of the item N2 onto the construct *behavioral intention* with a value of 0.712. Overall, these results indicate that the utilized measurement instrument satisfies both convergent and discriminant validity [Kli11].

Further tests for discriminant validity utilized in research on gamification (i. e., [KH14] and [HK15]) are based on the work by Pavlou et al. [PLX07], and can also be conducted using the results delivered by SmartPLS. Firstly, validity is indicated if the square root of the Average Variance Extracted (AVE) of each latent variable (i. e., the extent to which constructs explain the variances of their indicators) is larger than its cross-correlation with all other constructs. As the data in Table 8.10 shows, this is the case; specifically, the values on the diagonal of the matrix (the AVEs) are always larger than all values to the left or below (the correlations). Secondly, the AVE of all constructs should exceed 0.50, which suggests that they "[account] for a majority of the variance in [their]

Table 8.10: Field Experiment: User experience construct correlations and AVEs (bold).

Constructs	Usab.	Usef.	Pos. Em.	Neg. Em.	Beh. Int.
Usability	**0.844**				
Usefulness	0.599	**0.819**			
Pos. Emotions	0.319	0.164	**0.781**		
Neg. Emotions	-0.716	-0.515	-0.249	**0.887**	
Beh. Intention	0.540	0.629	0.324	-0.488	**0.852**

indicators on average" [MPP11, p. 313]. This is satisfied by the measurement instrument at hand, which exhibits AVEs between 0.610 for *positive emotions* and 0.787 for *usability*. In Table 8.10, these values can be obtained for each construct by multiplying the values on the diagonal with themselves. Lastly, further support for discriminant validity is given if no construct exhibits a high correlation with another latent variable. Referring once again to the data in Table 8.10 and employing 0.90 as a threshold [PLX07, Kli11], the result of this test are positive as well. In summary, the findings of the CFA and these three tests suggests that reasonable support for construct validity can be claimed, and thus meaningful conclusions can be gained from the survey data.

8.2.4 Model Quality Results

Initial insights into the impacts of gamification on the quality of process models can be gained from the descriptive statistics presented in Tables 8.11–8.14, which depict the mean and median model scores across quality metrics, quality characteristics, and overall model quality, respectively. Note that metrics for completeness are subdivided into two separate categories, namely those whose values students were instructed to consider while working on the case study, and those for which this was not the case. Unsurprisingly, the values of most quality metrics belonging to the second category (labeled *completeness 2*) are close to zero, and thus they will be disregarded in the following analyses. Accordingly, two overall quality scores are presented: one indicator incorporating all completeness metrics, and a second value excluding those of minor relevance. Finally, both tables contain separate columns for the winter terms 2015/16 and 2016/17 indicating for each quality metric whether it was shown to students (✔) or not (✖), i. e., whether it was configured as visible in the respective workspaces as described in Section 7.2.3. Looking at the mean values shown in Tables 8.11 and 8.12, the following observations can be made:

Readability metrics. For most readability metrics, almost no differences can be detected across most semesters independently of the presence of gamification and the mode of work (group vs. individual). This may be due to the fact that readability is a comparably trivial concept so that even novice modelers naturally uphold high quality values while modeling. For instance, results indicate that the fact that nodes should not overlap does not need to be taught, but is inherently clear to most. Nevertheless, some metrics seem to be more difficult to comprehend, and thus exhibit lower average scores, most notably angular resolution and orthogonality. In 2016, a notable increase of these two metrics can be observed, thereby also yielding an increase of the overall readability score. This effect persists independently of the presence of gamification, but is more pronounced in the presence of game design elements. This suggests that “soft goals” such as described in Section 8.2.1 have a significant impact on the effort that individuals expend to maintain model quality. Conversely, simply providing real-time quality feedback and gamification does not seem to suffice if the users of a modeling software are not supported in setting specific quality goals for themselves, which is consistent with the goal-setting theory outlined in Section 3.5.4.

Complexity metrics. Looking at metrics considering model complexity, more pronounced differences between individual semesters can be detected. Firstly, for the majority of metrics, results of 2013, 2014, and 2016 without gamification are on a similar level, although the latter exhibits a higher quality for, e. g., size and diameter. Secondly, independently of gamification, about half of the average measurements taken in 2015—specifically for the metrics size, diameter, connector mismatch, control flow complexity, cyclicity, and token split—have considerable lower values than the remaining observations. This can be attributed to changes in the course design, most notably the switch from group to individual work and the specification of an exact number of models to be created by course participants, which was not the case before. As a result, the average number of models per student was much lower in 2015, but the average size and overall complexity of these models increased. Finally, with the exception of the metrics average connector degree and maximum connector

Table 8.11: Field experiment: Descr. statistics for model quality (mean).

Metric	**2013/14**	**2014/15**	**2015/16**			**2016/17**		
			S	*Gamif.*	~~*Gamif.*~~	*S*	*Gamif.*	~~*Gamif.*~~
Sample Size N	425	435		277	273		560	343
Readability	0.8634	0.8656		0.8515	0.8463		0.9231	0.9128
Edge Crossings	0.9935	0.9951	✔	0.9977	0.9978	✔	0.9988	0.9978
Edge Bends	0.9969	0.9973	✔	0.9986	1.0000	✔	1.0000	1.0000
Node Occlusion	0.9977	0.9983	✔	0.9878	0.9957	✔	0.9781	0.9840
Angular Resolution	0.7442	0.7696	✔	0.7545	0.7423	✔	0.9188	0.8928
Consistent Flow 1	0.7565	0.7746	✖	0.7558	0.7384	✖	0.8315	0.8090
Consistent Flow 2	0.8924	0.8749	✔	0.8011	0.8000	✔	0.8941	0.8994
Orthogonality	0.6626	0.6495	✖	0.6652	0.6502	✔	0.8407	0.8068
Complexity	0.7200	0.7342		0.6459	0.6421		0.7849	0.7379
Size	0.7014	0.6678	✔	0.4173	0.4040	✔	0.8207	0.7697
Diameter	0.8111	0.7605	✔	0.6332	0.6117	✔	0.8820	0.9012
Density	0.9502	0.9630	✖	0.9695	0.9723	✖	0.9822	0.9795
Conn. Coefficient	0.9311	0.9476	✖	0.9183	0.9273	✔	0.9709	0.9607
Avg. Conn. Degree	0.8813	0.8921	✖	0.9311	0.9328	✖	0.7339	0.6943
Max. Conn. Degree	0.7535	0.7805	✖	0.7978	0.7866	✖	0.6304	0.5605
Conn. Mismatch	0.3334	0.3827	✖	0.2577	0.2630	✔	0.5170	0.4164
Control Flow Compl.	0.7691	0.7707	✔	0.7127	0.6986	✔	0.8510	0.7746
Cyclicity	0.8229	0.8282	✔	0.6351	0.6540	✔	0.8610	0.8382
Token Split	0.6273	0.6924	✖	0.4173	0.4388	✖	0.8354	0.7895
Sources and Sinks	0.3388	0.3908	✔	0.4152	0.3736	✔	0.5500	0.4329
Completeness	0.5301	0.5038		0.5543	0.4627		0.8639	0.7684
Short Names	0.9995	0.9985	✔	0.9991	0.9952	✔	1.0000	0.9997
Descriptions	0.4017	0.3029	✔	0.3230	0.2520	✔	0.9647	0.8962
Object Types	0.2349	0.2601	✔	0.3251	0.1679	✔	0.5595	0.4067
Roles	0.4842	0.4535	✔	0.5700	0.4357	✔	0.9315	0.7710
Completeness 2	0.0430	0.0418		0.0174	0.0127		0.0174	0.0184
Names	0.0197	0.0317	✖	0.0305	0.0181	✖	0.0868	0.0913
Notes	0.0049	0.0019	✖	0.0009	0.0001	✖	0.0031	0.0020
Business Rules	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0018	0.0029
Documents	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
KPIs	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0018	0.0029
Refinements	0.1824	0.1544	✖	0.0972	0.0918	✖	0.0713	0.0729
Resources	0.2233	0.2300	✖	0.0456	0.0149	✖	0.0042	0.0036
Risks	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0018	0.0029
Services	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0018	0.0029
System Components	0.0000	0.0000	✖	0.0000	0.0024	✖	0.0018	0.0029

Table 8.12: Field experiment: Continuation of descriptive statistics for model quality (mean).

Metric	**2013/14**	**2014/15**	**2015/16**			**2016/17**		
			S	*Gamif.*	~~*Gamif.*~~	*S*	*Gamif.*	~~*Gamif.*~~
Sample Size N	425	435		277	273		560	343
Overall Quality	0.5885	0.5914		0.5561	0.5432		0.6558	0.6278
w/o Completeness 2	0.7045	0.7012		0.6839	0.6504		0.8573	0.8064

degree, the values of all metrics in 2016 in the presence of gamification strongly exceed those of all other datasets. Once again, this points to the importance of "soft goals" to enable an effective operationalization of gamified quality feedback. Consequently, results of 2015 indicate that students were either unable to leverage gamification, or simply decided not to do so in the absence of specific instructions to do so.

Completeness. For the most important completeness metrics, which are short names, descriptions, object types, and roles, the following observations can be made. Firstly, short names have been provided for almost all model elements, as this is the first action whenever a new element is created. Secondly, as comparisons of the values of 2015 and 2016 reveal, gamification has a strong impact on model completeness, and thus the values in the presence of game design elements are consistently higher. Specifically, the increase in average completeness is 9% in 2015, and 10% in 2016. Lastly, as for readability and completeness, modelers expend more effort for the improvement of model quality if this requirement is clearly communicated. While this effect is most pronounced for the gamified HBM, it still persists with a large effect even when gamification is not present. Overall, the completeness values of some metrics are roughly 2-3 times as high in 2016 with gamification as the averages of the remaining semesters. Due to their overall irrelevance, no further discussion of metrics belonging to the category *completeness 2* is provided.

Overall quality. The averages of all quality characteristics indicate that overall, no significant differences exist in the results of the case studies conducted in the winter terms 2013 and 2014. In 2015, an increase in the difficulty of maintaining high quality levels could be observed, which can be attributed to the organizational changes implemented in the lecture. Finally, as expected from the intermediary results for readability, complexity, and completeness, the overall average model quality in 2016 exceeds that of the remaining instantiations of the lecture. These effects can be observed regardless of the inclusion of metrics belonging to the category *completeness 2*, but are more pronounced in light of their exclusion. On the aggregated level, the impact of gamification can be quantified at 3% in 2015, and 5% in 2016. While this may seem small at first, the effects may increase even further in real-world modeling projects involving large and complex business processes if the influence of gamification is statistically significant. In such a scenario, the readability, complexity, and completeness of models become less trivial to maintain, and thus the quality gap between models created with and without gamification will increase.

The mean is a good measure of centrality for samples with symmetric empirical distributions, i. e., when an identifiable middle point exists such as in a normal distribution. However, for asymmetric (i. e., skewed) distributions, distributions with more than one center, or samples containing many extremal values that cannot be excluded as outliers, the mean may be inappropriate in some cases and the median should be preferred [Fie09, Pag13]. In the dataset at hand, the empirical distributions of most quality metrics differ significantly from a normal distribution across all semesters and independently of gamification. Furthermore, for many quality metrics, a high frequency of values close to 0.0 or 1.0 can be observed, with values in the center of the range occurring more sparsely. This suggests that the median may be a more appropriate measure of central tendency for model quality than the mean, and thus complementary data is provided in Tables 8.13 and 8.14. However, the main conclusions drawn from the average quality scores are still valid for median values, and thus no separate discussion of these tables is conducted. However, it should be noted that the differences between the gamified and non-gamified versions of the HBM become more pronounced for some metrics, whereas they disappear for others when the median is employed.

Having presented descriptive statistics, the next step lies in examining whether the impacts of gamification on model quality are statistically significant. To that extent, three tests are carried out: 1. Examination of the impacts of gamification on model quality for the dataset obtained from the winter term 2015/16. 2. Examination of the impacts of gamification on model quality for the dataset obtained from the winter term 2016/17. 3. Examination of the impacts of “soft

Table 8.13: Field experiment: Descr. statistics for model quality (median).

Metric	**2013/14**	**2014/15**	**2015/16**			**2016/17**		
			S	*Gamif.*	~~*Gamif.*~~	*S*	*Gamif.*	~~*Gamif.*~~
Sample Size N	425	435		277	273		560	343
Readability	0.8690	0.8686		0.8667	0.8573		0.9349	0.9186
Edge Crossings	1.0000	1.0000	✔	1.0000	1.0000	✔	1.0000	1.0000
Edge Bends	1.0000	1.0000	✔	1.0000	1.0000	✔	1.0000	1.0000
Node Occlusion	1.0000	1.0000	✔	1.0000	1.0000	✔	1.0000	1.0000
Angular Resolution	0.7500	0.7620	✔	0.7690	0.7500	✔	0.9470	0.9170
Consistent Flow 1	0.7500	0.7890	✖	0.7500	0.7390	✖	0.8500	0.8330
Consistent Flow 2	1.0000	1.0000	✔	0.8460	0.8240	✔	0.9440	0.9430
Orthogonality	0.6920	0.6670	✖	0.6960	0.6670	✔	0.8820	0.8330
Complexity	0.7520	0.7506		0.6415	0.6445		0.7821	0.7454
Size	0.9000	0.8000	✔	0.4000	0.3000	✔	1.0000	1.0000
Diameter	0.9000	0.9000	✔	0.7000	0.7000	✔	1.0000	1.0000
Density	0.9740	0.9800	✖	0.9760	0.9810	✖	0.9880	0.9860
Conn. Coefficient	1.0000	1.0000	✖	0.9640	0.9580	✔	1.0000	1.0000
Avg. Conn. Degree	1.0000	1.0000	✖	1.0000	0.9640	✖	0.7500	0.7500
Max. Conn. Degree	1.0000	1.0000	✖	1.0000	0.7500	✖	0.7500	0.5000
Conn. Mismatch	0.0000	0.2500	✖	0.1430	0.1430	✔	0.5000	0.3330
Control Flow Compl.	1.0000	1.0000	✔	0.8570	0.8570	✔	1.0000	1.0000
Cyclicity	1.0000	1.0000	✔	0.8060	0.8180	✔	1.0000	1.0000
Token Split	0.6000	1.0000	✖	0.2000	0.2000	✖	1.0000	1.0000
Sources and Sinks	0.0000	0.0000	✔	0.0000	0.0000	✔	0.5000	0.5000
Completeness	0.5355	0.5000		0.5700	0.4573		0.9038	0.7750
Short Names	1.0000	1.0000	✔	1.0000	1.0000	✔	1.0000	1.0000
Descriptions	0.4500	0.3120	✔	0.3950	0.2250	✔	1.0000	1.0000
Object Types	0.0000	0.0000	✔	0.0830	0.0000	✔	0.6670	0.2860
Roles	0.5000	0.2000	✔	0.8000	0.3330	✔	1.0000	1.0000
Completeness 2	0.0250	0.0200		0.0000	0.0000		0.0000	0.0000
Names	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Notes	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Business Rules	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Documents	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
KPIs	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Refinements	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Resources	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Risks	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
Services	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000
System Components	0.0000	0.0000	✖	0.0000	0.0000	✖	0.0000	0.0000

Table 8.14: Field experiment: Continuation of descriptive statistics for model quality (median).

Metric	**2013/14**	**2014/15**	**2015/16**			**2016/17**		
			S	*Gamif.*	*~~Gamif.~~*	*S*	*Gamif.*	*~~Gamif.~~*
Sample Size N	425	435		277	273		560	343
Overall Quality	0.5978	0.5952		0.5722	0.5447		0.6568	0.6323
w/o Completeness 2	0.7161	0.6995		0.6923	0.6456		0.8612	0.8137

goals” on the capability of modelers to operationalize game design elements by comparing the data of 2015/16 and 2016/17 with gamification. Data from 2013 and 2014 is disregarded, as the differences in course organization and the utilized versions of the HBM would make the validity of any conclusions drawn from a comparison with more recent data highly questionable.

Gamification vs. ~~Gamification~~ (2015). A test for normality reveals that the data of most quality metrics is non-normally distributed for both the experimental group with gamification and the control group without. Therefore, a two-sided *Mann-Whitney-U* test was conducted to check for statistically significant differences in the distributions of both groups. The results of this test are summarized in Table 8.15 according to the scheme presented in Section 8.1. Results confirm the observation made above that the impacts of gamification on readability and complexity are negligible. In case of those metrics of both types with high average values, i. e., 0.9 or above, this may be due to the fact that most of the models created by students were comparably simple, and thus maintaining high quality levels was possible without difficulty. For the remaining metrics, including angular resolution, size, and cyclicity, students may either not have felt the need to ensure a high model quality in the the absence of specific instructions to do so, or were incapable of doing so due to insufficient support and their unawareness of the help features described in Section 7.4.3. For all relevant completeness metrics besides short names, results allow concluding with a very low error probability that the models of both groups do not belong to the statistical population. Referring back to the descriptive statistics presented above, this means that these metrics are likely to exhibit higher values when gamification is present. Effect sizes for these three metrics range from small to medium. On an aggregated level, students working with the gamified version of the HBM have created models with a higher completeness and overall quality with medium and small effect sizes respectively. In summary, the only statistically significant impact of gamification in the winter term 2015/16 can be observed for model completeness. This may be due to the fact that students were explicitly instructed to, e. g., create object models and roles, reference those models in their business processes, and create description texts where appropriate.

Gamification vs. ~~Gamification~~ (2016). As before, neither dataset conforms to a normal distribution, and thus a two-sided *Mann-Whitney-U* test was conducted to check for statistically significant differences in the distributions of both groups. The results of this test are summarized in Table 8.16 according to the scheme presented in Section 8.1. Results confirm the observation made above that the impacts of gamification on model quality were more pronounced in 2016 than in 2015. Specifically, statistically significant differences at the alpha levels 0.05 or 0.01 can be detected for almost all quality metrics, with some notable exceptions being the tree “trivial” readability metrics edge crossings, edge bends, and node occlusion. However, it should be noted that the effect sizes are moderate in most cases, and range from as low as 0.0602 to as high as 0.2657. These results do not only hold for quality metrics for which students received actual quality feedback, but also for some that were disabled resp. hidden. This may either be the result of a positive correlation between the affected hidden and some of the visible metrics, or due to differences in the modeling behavior of students caused by gamification that manifest themselves in all areas of model quality. On an aggregated level, students working with the gamified version of the HBM have created models with higher readability and completeness, and lower complexity. Ultimately, this results in the observation of a highly significant, large impact of gamification on overall model quality. In summary, the results of the winter term 2016/17 confirm that gamification can have a significant impact on the quality of process models created by modeling novices.

Gamification (2015 vs. 2016). As the descriptive statistics presented at the beginning of this section indicate, variations in the use of gamification between 2015/16 and 2016/17 have led to considerable differences in the quality of the resulting models. Since the implementation of the gamification module used in both semesters was identical, the reasons for these differences must be sought

Table 8.15: Field experiment: Model quality test: Gamification vs. ~~Gamification~~ (2015).

Metric	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
Readability Shown	✔	36784.0	-0.551	0.582	-	-	-
Edge Crossings	✔	36297.5	-1.171	0.242	-	-	-
Edge Bends	✖	37674.0	-0.993	0.321	-	-	-
Node Occlusion	✖	37406.0	-0.825	0.409	-	-	-
Angular Resolution	✔	36011.0	-0.966	0.334	-	-	-
Consistent Flow 2	✔	37495.0	-0.172	0.864	-	-	-
Readability Hidden	✔	36013.0	-0.965	0.335	-	-	-
Consistent Flow 1	✔	36249.5	-0.838	0.402	-	-	-
Orthogonality	✔	36408.0	-0.753	0.451	-	-	-
Complexity Shown	✔	36388.0	-0.763	0.445	-	-	-
Size	✔	36923.0	-0.484	0.628	-	-	-
Diameter	✔	35900.5	-1.041	0.298	-	-	-
Control Flow Compl.	✔	36006.0	-1.003	0.316	-	-	-
Cyclicity	✔	36670.0	-0.649	0.516	-	-	-
Sources and Sinks	✔	36240.0	-0.996	0.319	-	-	-
Complexity Hidden	✔	37336.5	-0.254	0.799	-	-	-
Conn. Coefficient	✖	37658.0	-0.085	0.932	-	-	-
Conn. Mismatch	✔	37298.5	-0.290	0.771	-	-	-
Density	✔	37356.5	-0.246	0.806	-	-	-
Avg. Conn. Degree	✔	37806.5	-0.002	0.998	-	-	-
Max. Conn. Degree	✔	36631.0	-0.685	0.493	-	-	-
Token Split	✔	36919.0	-0.501	0.616	-	-	-
Completeness Shown	✖	27692.5	-5.434	0.000	***	0.2317	M
Descriptions	✔	31045.5	-3.646	0.000	***	0.1555	S
Object Types	✖	29789.0	-4.764	0.000	***	0.2031	M
Roles	✔	31644.5	-3.515	0.000	***	0.1499	S
Short Names	✖	37517.0	-0.681	0.496	-	-	-
Overall Quality	✖	30056.0	-4.161	0.000	***	0.1774	S
Readability	✔	35983.5	-0.980	0.327	-	-	-
Complexity	✔	37168.0	-0.345	0.730	-	-	-
Completeness	✖	27692.5	-5.434	0.000	***	0.2317	M

Table 8.16: Field experiment: Model quality test: Gamification vs. ~~Gamification~~ (2016).

Metric	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
Readability Shown	✔	86122.0	-2.614	0.009	***	0.0870	X
Edge Crossings	✖	93802.0	-1.161	0.246	-	-	-
Edge Bends	✔	96040.0	0.000	1.000	-	-	-
Node Occlusion	✔	95527.0	-0.484	0.628	-	-	-
Angular Resolution	✔	83124.0	-3.522	0.000	***	0.1172	S
Consistent Flow 2	✔	94736.5	-0.359	0.720	-	-	-
Orthogonality	✔	82838.0	-3.510	0.000	***	0.1168	S
Readability Hidden	✔	86597.5	-2.490	0.013	**	0.0829	X
Consistent Flow 1	✔	86597.5	-2.490	0.013	**	0.0829	X
Complexity Shown	✔	79278.5	-4.409	0.000	***	0.1467	S
Size	✖	89490.0	-1.909	0.056	*	0.0635	X
Diameter	✔	92233.5	-1.239	0.215	-	-	-
Conn. Coefficient	✖	87474.0	-2.521	0.012	**	0.0839	X
Conn. Mismatch	✔	83188.5	-3.555	0.000	***	0.1183	S
Control Flow Compl.	✖	85409.5	-3.116	0.002	***	0.1037	S
Cyclicity	✔	90805.0	-1.604	0.109	-	-	-
Sources and Sinks	✔	81746.0	-3.999	0.000	***	0.1331	S
Complexity Hidden	✔	81733.5	-3.768	0.000	***	0.1254	S
Density	✔	90182.0	-1.573	0.116	-	-	-
Avg. Conn. Degree	✔	83291.5	-3.435	0.001	***	0.1143	S
Max. Conn. Degree	✔	81761.5	-3.925	0.000	***	0.1306	S
Token Split	✖	88206.0	-2.279	0.023	**	0.0759	X
Completeness Shown	✖	66536.5	-7.868	0.000	***	0.2618	M
Descriptions	✖	74804.5	-7.984	0.000	***	0.2657	M
Object Types	✔	76016.5	-5.435	0.000	***	0.1809	S
Roles	✖	78052.5	-7.134	0.000	***	0.2374	M
Short Names	✖	95480.0	-1.808	0.071	*	0.0602	X
Overall Quality	✔	61336.0	-9.123	0.000	***	0.3036	L
Readability	✔	84395.0	-3.063	0.002	***	0.1019	S
Complexity	✔	77671.0	-4.830	0.000	***	0.1607	S
Completeness	✖	66536.5	-7.868	0.000	***	0.2618	M

in contextual factors and factors relating to how the modeling case study was conducted. Generally speaking, there is no reason to suspect that the students in both semesters do not belong to the same statistical population, and thus any differences are unlikely to be attributable to the change in participants. Furthermore, the task description of the case study (see Section C.1) remained identical with the exception of minor details, and thus all students had to accomplish the same tasks. Thus, the main difference between both datasets lies in the hard and soft goals communicated to students as described in Section 8.2.1. In particular, the importance of utilizing the quality feedback provided by the gamification module was stressed as part of the latter.

Table 8.17 provides a summary of the mean and median quality values on the metric, characteristic, and model level. Note that the overall quality depicted here differs from Table 8.12 and Table 8.14 due to the exclusion of hidden metrics. As these descriptive statistics show, the changes administered to the organization of the case study have resulted in considerable quality improvements for complexity and completeness, and the readability metrics angular resolution and consistent flow 2 (left-to-right). These improvements are also reflected in overall model quality, which exhibits an increase of approximately 20% regardless of the utilized measure of centrality.

Once again, a test for normality shows that the data of most quality metrics is non-normally distributed for both the experimental group with "soft goals" and the control group without. Therefore, a two-sided *Mann-Whitney-U* test was conducted to check for statistically significant differences in the distributions of both groups. The results of this test are summarized in Table 8.18 according to the scheme presented in Section 8.1. Results confirm the previous observation that the impacts of gamification on model quality were more pronounced in 2016 than in 2015. Statistically significant differences at the alpha level 0.01 can be detected for almost all quality metrics with the exception of the two "trivial" readability metrics edge bends, and node occlusion. Interestingly, the test also reveals a significant difference for the metric edge crossings despite the minor differences in means and medians between 2015 and 2016. On the aggregated level of quality characteristics, all differences are also statistically significant at the alpha level 0.01, and the effect sizes can consistently be described as large. This effect also persists at the level of overall model quality. In summary, the results show that the application of gamification was much more successful in

Table 8.17: Field experiment: Descriptive statistics for model quality (gamification).

Metric	2015 Mean	2016 Mean	Diff.	2015 Median	2016 Median	Diff.
Readability Shown	0.9079	0.9580	▲0.0501	0.9200	0.9700	▲0.0500
Edge Crossings	0.9977	0.9988	▲0.0011	1.0000	1.0000	▲0.0000
Edge Bends	0.9986	1.0000	▲0.0014	1.0000	1.0000	▲0.0000
Node Occlusion	0.9878	0.9781	▼-0.0097	1.0000	1.0000	▲0.0000
Angular Resolution	0.7545	0.9188	▲0.1643	0.7690	0.9470	▲0.1780
Consistent Flow 2	0.8011	0.8941	▲0.0930	0.8460	0.9440	▲0.0980
Complexity Shown	0.5627	0.7929	▲0.2302	0.5644	0.8000	▲0.2356
Size	0.4173	0.8207	▲0.4034	0.4000	1.0000	▲0.6000
Diameter	0.6332	0.8820	▲0.2488	0.7000	1.0000	▲0.3000
Control Flow Compl.	0.7127	0.8510	▲0.1383	0.8570	1.0000	▲0.1430
Cyclicity	0.6351	0.8610	▲0.2259	0.8060	1.0000	▲0.1940
Sources and Sinks	0.4152	0.5500	▲0.1348	0.0000	0.5000	▲0.5000
Completeness Shown	0.5543	0.8639	▲0.3096	0.5700	0.9038	▲0.3338
Descriptions	0.3230	0.9647	▲0.6417	0.3950	1.0000	▲0.6050
Object Types	0.3251	0.5595	▲0.2344	0.0830	0.6670	▲0.5840
Roles	0.5700	0.9315	▲0.3615	0.8000	1.0000	▲0.2000
Short Names	0.9991	1.0000	▲0.0009	1.0000	1.0000	▲0.0000
Overall Quality Shown	0.6750	0.8716	▲0.1966	0.6739	0.8824	▲0.2085

Table 8.18: Field experiment: Model quality test: Gamification (2015 vs. 2016).

Metric	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
Readability Shown	✖	35905.0	-12.703	0.000	***	0.4391	L
Edge Crossings	✖	70909.5	-3.652	0.000	***	0.1262	S
Edge Bends	✖	77280.0	-1.422	0.155	-	-	-
Node Occlusion	✖	77541.0	-0.020	0.984	-	-	-
Angular Resolution	✖	27007.5	-15.640	0.000	***	0.5406	L
Consistent Flow 2	✖	53716.5	-7.473	0.000	***	0.2583	M
Complexity Shown	✖	28029.5	-15.058	0.000	***	0.5205	L
Size	✖	32103.5	-14.492	0.000	***	0.5009	L
Diameter	✖	30802.5	-15.162	0.000	***	0.5241	L
Control Flow Compl.	✖	59742.0	-5.940	0.000	***	0.2053	M
Cyclicity	✖	55432.5	-7.586	0.000	***	0.2622	M
Sources and Sinks	✖	65457.0	-3.987	0.000	***	0.1378	S
Completeness Shown	✖	17852.5	-18.288	0.000	***	0.6321	L
Descriptions	✖	3265.0	-25.390	0.000	***	0.8776	L
Object Types	✖	54142.5	-7.376	0.000	***	0.2549	M
Roles	✖	40616.0	-14.564	0.000	***	0.5034	L
Short Names	✖	76440.0	-2.849	0.004	***	0.0985	X
Overall Quality Shown	✖	10225.0	-20.459	0.000	***	0.7072	L

2016 with regard to its impacts on model quality at all levels. This indicates that the presence of game design elements in an application may not be sufficient for their effective operationalization, and that further measures are sometimes required to ensure the latter. Exceptions may be found in gamified applications such as those presented in Section 3.2.1, where gamification is a central component of the user experience rather than a supporting feature.

8.2.5 System Use Results

Initial insights into the impacts of gamification on the usage behavior of students working on the case study with the HBM can be gained from the descriptive statistics presented in Table 8.19. Here, the means, standard deviations (SD), and medians of the event counts calculated as described in Section 8.2.2 are depicted. Additionally, the final column indicates the relative increase or decrease of the median number of events across users in the presence of gamification. For instance, the value 0.18 in the first line is calculated as $(1473 - 1211)/1473$ and represents an 18%-increase in the overall number of events fired by users of the gamified version of the HBM. The median was chosen as a measure of centrality for the comparison rather than the mean as most variables are non-normally distributed and are influenced by extremal values. Note that Table 8.19 only lists a subset of the set of events for which data was collected, specifically those events for which statistical testing yielded significant results (see Table 8.20 below) and their respective parent nodes in the event hierarchy. More detailed results for the full set of events are provided in Section C.3 of the appendix.

As before, the data of event types is non-normally distributed for both the experimental group with gamification and the control group without. Therefore, a one-sided *Mann-Whitney-U* test was conducted to check for statistically significant differences in the distributions of both groups. The results of this test are summarized in Table 8.20 according to the scheme presented in Section 8.1. Together with the descriptive statistics presented in Table 8.19, the following observations can be made:

- **Modeling sessions**. Results indicate that students having worked with the gamified version of the HBM exhibited higher levels of engagement and expended more effort on the case study. Specifically, the values of *events.user.login* and *Session length sum* indicate that they conducted 30% more modeling sessions and spent 34% more time using the software. This may also be one of the underlying reasons for the higher model quality achieved by the experimental group. The results for both indicators are statistically significant at the alpha levels 0.05 and 0.1, and exhibit small effect sizes, respectively. No significant effects could be observed for both definitions of the *average modeling session length*, which indicates that students *only* modeled more often, but not longer at a time. Similarly, gamification did not result in any increase or decrease of the speed with which the tool was used, and thus the *average decision time* remains unchanged.

- **Model interaction**. Users of the gamified version of the HBM have fired more events related to the interaction with models for all model types,

Table 8.19: Field experiment: Descriptive statistics for system use.

	Gamification			~~**Gamification**~~			
Event Type	**Mean**	**SD**	**Med.**	**Mean**	**SD**	**Med.**	**Diff.**
events	1597.76	948.34	1473	1496.05	990.54	1211	▲0.18
editor	1527.19	921.87	1417	1462.07	975.08	1191	▲0.16
modelClosed	57.75	35.81	49	41.37	23.82	34	▲0.31
modelOpened	61.77	36.87	56	44.14	24.44	36	▲0.36
modelSaved	34.34	20.9	29	27.72	17.81	26	▲0.1
petriNet	1373.33	861.96	1294	1348.8	922.96	1100	▲0.15
created	357.75	228.95	322	380.81	295.62	310	▲0.04
graphics	4.28	14.78	0	6.3	23.71	0	=
note	0.47	3.13	0	0.56	1.69	0	=
rectangleRound	0.06	0.56	0	0.47	1.78	0	=
deleted	9.58	14.67	5	11.95	18.95	4	▲0.2
model.place	1.67	2.97	0	2.56	3.64	1	▼1
propertyChanged	1006	633.32	936	956.07	643.56	787	▲0.16
graphics	706.14	450.17	618	693.58	474.15	565	▲0.09
arrangeNodes	1.06	2.13	0	0.53	1.1	0	=
subdiagram	4.78	4.45	3	5.56	4.15	5	▼0.4
text	2.34	9.33	0	5.19	22.52	0	=
properties	284.46	186.81	246	253.49	180.67	204	▲0.17
General	241.68	157.34	224	209.05	151.47	176	▲0.21
Node	241.42	157.44	224	208.7	151.38	176	▲0.21
Description	72.37	60.84	67	48.95	63.97	30	▲0.55
Notes	0.29	1.03	0	0.28	1.83	0	=
references	15.41	14.55	13	9	12.42	5	▲0.62
entityType	7.27	9.67	2	4.05	7.48	0	▲1
resource	0.78	2.54	0	0	0	0	=
role	7.27	7.58	5	4.91	6.87	2	▲0.6
projectManager	18.24	8.23	17	15.84	7.75	14	▲0.18
modelCreated	12.91	4.98	13	11.72	5.39	11	▲0.15
quality.changed	25.75	19.09	22	4.88	14.32	0	▲1
*reward.**	14.46	8.74	13	4.09	3.96	3	▲0.77
user	11.05	5.86	10	8.58	4.34	8	▲0.2
login	10.56	5.7	10	8.3	4.32	7	▲0.3
profile.changed	0.43	0.52	0	0.26	0.44	0	=
Event Types	55	8.12	54	49.83	7.17	49	▲0.09
Session Length	-	-	-	-	-	-	-
Sum	106868	81762	96042	97989	102478	63245	▲0.34
Avg w/ Empty	11974	12115	8538	12919	19653	7177	▲0.16
Avg w/o Empty	14093	13019	10174	15939	21007	9854	▲0.03
Avg. Decision Time	84.52	96.29	50.78	77.23	83.3	50.84	=

in particular concerning the creation of models (▲15%), opening models (▲31%), closing models (▲31%), and saving models (▲10%). Results for all four event types are significant, although only at the alpha level 0.1 for save operations and model creation, and the respective effect sizes range from small to medium. Overall, this may be interpreted as an extension of the results concerning modeling sessions, and points to higher levels of engagement and effort related to gamification. The higher occurrence of save operations may further be a consequence of the reward system, which distributes points to modelers whenever a model is saved.

- **Model completeness**. As already confirmed by the the data on model quality, students using gamification have created more object models (also called entity types) and roles, and linked them to the respective elements of their process models. Consequently, a higher occurrence of related events can also be observed, with the excess ranging from 0% to 100% as measured by the differences in medians. Besides object types and roles, this also includes resource models, which were not part of the case study requirements communicated to students. For all three event types, the results are statistically significant at the alpha levels 0.05 and 0.01, and the related effects are of medium size. Similarly, the impact of gamification on events related to the modification of description texts impacts the median of 55%, and this difference is highly significant with a medium effect size.

- **Gamification events**. Users of the gamified HBM have fired considerably more events related to gamification than the control group, which can reasonably be expected. This concerns all events matching the expression *events.reward.** as well *quality.changed* events, for which the differences in medians correspond to 77% and 100% respectively. In both cases, the Mann-Whitney-U test yields results at the significance level 0.01 with large effect sizes.

- **Diverse use**. Results of the aggregated *event types* measure indicate that students working with gamification have used a larger portion of the functionality of the HBM that is tracked by corresponding events. Specifically, this increase is measured at approximately 9%, and the corresponding test reveals a high significance of this difference and a medium (large, if rounded to two digits after the decimal point) effect size. This effect of gamification is also reflected in the observations that can be made for certain other events, which have little explanatory power by themselves, but together point to a more diverse use of the software connected to gamification. Specifically, this concerns the events with the suffixes *model.place*, *note*, *rounded rectangle*, *arrangeNodes*, *subdiagram*, *text*, and *Notes*.

- **Profile completion**. A small effect of gamification can also be reported for the completion of profile information by users, which indicates that the affected students were more inclined to disclose personal information about themselves to be displayed on their profile pages. The size of this effect may have been limited due to the fact that the current implementation does not provide a searchable registry of users, and only makes profiles accessible via the leaderboard.

In summary, the results show that the integration of game design elements into a business process modeling software can indeed have a significant impact on the behavior of its users. Specifically, the analyzed data points to an increased and more diverse use of the system by course participants belonging to the experimental group. Additionally, results confirm the findings about model quality presented in the previous section.

8.2.6 User Experience Results

Initial insights into the impacts of gamification on the user experience of students working on the case study can be gained from the descriptive statistics presented in Table 8.21. Here, the means, standard deviations (SD), and medians of the construct scores collected by means of the adapted meCUE questionnaire are depicted. For all constructs except the overall grade, possible values for most columns range from the minimum value 1 to the maximum value 7. Additionally, the final column indicates the relative increase or decrease of the average construct score in the presence of gamification. For instance, the value 0.13 in the first line is calculated as $(4.92 - 4.29)/4.92$ and represents a 13%-increase of the perceived usability of the gamified version of the HBM.

Table 8.20: Field experiment: System use hypothesis test results.

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
events	✔	1556.0	-0.7637	0.2237	-	-	-
editor	✔	1589.5	-0.5841	0.2807	-	-	-
modelClosed	✖	1210.0	-2.6183	0.0043	***	0.2370	M
modelOpened	✖	1183.0	-2.7630	0.0027	***	0.2501	M
modelSaved	✔	1362.0	-1.8043	0.0356	**	0.1634	S
petriNet	✔	1635.0	-0.3403	0.3678	-	-	-
created	✔	1697.5	-0.0054	0.4984	-	-	-
graphics	✔	1519.0	-1.2172	0.1139	-	-	-
note	✔	1569.0	-1.6175	0.0929	*	0.1464	S
rectangleRound	✖	1600.5	-1.7024	0.0695	*	0.1541	S
deleted	✔	1636.0	-0.3378	0.3686	-	-	-
model.place	✔	1459.5	-1.3924	0.0830	*	0.1261	S
propertyChanged	✔	1594.5	-0.5573	0.2898	-	-	-
graphics	✔	1630.5	-0.3644	0.3588	-	-	-
arrangeNodes	✔	1462.5	-1.4531	0.0724	*	0.1316	S
subdiagram	✔	1452.5	-1.3290	0.0926	*	0.1203	S
text	✔	1397.0	-2.1938	0.0160	**	0.1986	S
properties	✔	1497.0	-1.0799	0.1410	-	-	-
General	✔	1452.0	-1.3210	0.0939	*	0.1196	S
Node	✔	1450.0	-1.3318	0.0921	*	0.1206	S
Description	✔	1196.0	-2.6974	0.0033	***	0.2442	M
Notes	✔	1549.0	-1.6843	0.0747	*	0.1525	S
references	✔	1116.5	-3.1310	0.0008	***	0.2835	M
entityType	✖	1295.5	-2.2730	0.0110	**	0.2058	M
resource	✖	1419.0	-2.7970	0.0024	***	0.2532	M
role	✔	1269.0	-2.3291	0.0097	***	0.2109	M
projectManager	✔	1314.5	-2.0609	0.0195	**	0.1866	S
modelCreated	✔	1373.5	-1.7463	0.0405	**	0.1581	S
quality.changed	✖	389.5	-7.1526	0.0000	***	0.6476	L
*reward.**	✖	355.0	-7.2167	0.0000	***	0.6534	L
user	✔	1276.0	-2.2688	0.0115	**	0.2054	M
login	✔	1305.5	-2.1110	0.0173	**	0.1911	S
profile.changed	✖	1418.0	-1.8032	0.0474	**	0.1633	S
Event Types	✔	1085.0	-3.2921	0.0004	***	0.2980	M
Session Length							
Sum	✔	1414.0	-1.3339	0.0919	*	0.1208	S
Avg w/ Empty	✖	1598.0	-0.5386	0.2967	-	-	-
Avg w/o Empty	✖	1634.0	-0.3457	0.3663	-	-	-
Avg. Decision Time	✔	1589.0	-0.5868	0.2803	-	-	-

Table 8.21: Field experiment: Descriptive statistics for user experience.

	Gamification				~~**Gamification**~~				
Event Type	**n**	**Mean**	**SD**	**Med.**	**n**	**Mean**	**SD**	**Med.**	**Diff.**
Usability	51	4.92	1.19	5.00	28	4.29	1.34	4.83	▲0.13
Usefulness	50	5.09	1.08	5.33	28	4.94	0.99	5.00	▲0.03
Aesthetics	51	4.39	1.43	5.00	28	3.89	1.62	4.00	▲0.11
Negative Emotions	51	3.27	1.11	3.25	28	4.02	1.26	4.25	▼0.19
Positive Emotions	49	3.34	0.97	3.67	28	2.95	1.24	3.00	▲0.12
Behavioral Intention	51	4.16	1.25	4.40	28	3.87	1.33	4.10	▲0.07
Horus Grade	51	2.21	0.58	2.00	28	2.69	0.88	2.70	▼0.18

The presented values indicate that the students attribute the HBM with good usability and usefulness, and have a neutral view on its visual aesthetics. Furthermore, the software evoked rather low levels of emotional responses (both positive and negative), which may be due to its predominant role as a utilitarian rather than a hedonic IS. Additionally, students express moderate intentions regarding the continued use of the HBM, but give the tool an overall satisfactory school grade. The differences in the the values of both subsamples illustrate that the incorporation of game design elements leads to better average scores for all constructs, which is reflected in an increase of all values except *negative emotions* and the overall *Horus grade*, where lower grades are superior. Overall, the strongest impacts of gamification can be observed for the perceived usability of the HBM, the reduction of any negative emotions felt towards the tool, and its overall assessment by means of a school/university grade.

Testing for normality shows that the data of most user experience constructs is not consistently normally distributed for either subsample. Therefore, a one-sided *Mann-Whitney-U* test was conducted to check for statistically significant differences in the distributions of both groups. The results of this test are summarized in Table 8.22 according to the scheme presented in Section 8.1. Results confirm the observation that gamification has a highly significant impact on the perceived usability of the software, but not on its usefulness, i. e., it reduces the effort that is required for the use of the HBM, but does not make it more effective for the modeling tasks at hand. Furthermore, the interface elements through which gamification is presented to users make the HBM visually more pleasing as evidenced by an increase in its perceived aesthetics. However, it should be noted that the corresponding null hypothesis can only be rejected at a significance level 0.1, which is less stringent than the threshold of 0.05 commonly used in statistical testing. Accepting the alternative hypothesis under these conditions would result in a 10%-chance to commit a *Type I error*, i. e., to presume the existence of an effect in the population that is not actually present [Fie09]. Additionally, the results indicate that gamification has resulted in a statistically significant reduction of any negative emotions felt by students toward the HBM at the alpha level 0.01, and inversely increased positive emotions at the alpha level 0.1. This may either be a consequence of the increases in the perceived usability of the software, or could also indicate that executing serious activities in a gameful manner can compensate for potential frustrations that arise while doing so. Looking at the reported behavioral intentions of the participating students with regard to the HBM, no differences between the experimental group and the control group could be detected. Besides the mere ineffectiveness of game design elements in this context, this may also be partly attributed to the curiosity of first-semester students and their desire to be taught in the use of a variety of business process modeling languages and tools. Lastly, a highly significant effect of gamification can also be reported with regard to the overall assessment of the HBM by means of a school grade. In summary, the most important findings of this test can be summarized as follows: gamification reduces the effort required for using the HBM to work on a business process modeling task, it reduces negative emotions felt towards the software by its users, and it improves its overall assessment in terms of a single quantitative indicator. For these three constructs, the effect sizes reported in Table 8.22 also exceed the threshold of 0.2 presented in Section 8.2.1, which constitutes a medium-sized effect.

8.3 Laboratory Experiment

In this section, the results of a laboratory experiment carried out in the context of this research are reported. To that extent, Section 8.3.1 first describes the experiment itself, including its context and participants. Section 8.3.2 then outlines the dataset collected during the experiment, and the reliability and validity of this dataset is examined in Section 8.3.3. The section ends with the analysis of the obtained data and the presentation of results in Section 8.3.4.

Table 8.22: Field experiment: User experience hypothesis test results.

Construct	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
Usability	✖	540.5	-1.793	0.037	**	0.2017	M
Usefulness	✔	616.5	-0.875	0.193	-	-	-
Aesthetics	✔	578.0	-1.423	0.078	*	0.1601	S
Negative Emotions	✖	468.0	-2.527	0.005	***	0.2843	M
Positive Emotions	✔	496.0	-1.585	0.057	*	0.1830	S
Behavioral Intention	✔	644.0	-0.719	0.238	-	-	-
Horus Grade	✖	470.5	-2.533	0.005	***	0.2850	M

8.3.1 Description

To confirm the results of the field experiment and also gain additional insights about the motivational impacts of gamification for process modeling, a laboratory experiment was conducted subsequently to the completion of the lecture *Introduction to IS* in the winter term 2015/16. Participants were recruited as volunteers from the students having attended the lecture, and received €20 as a monetary compensation for their time and effort. The experiment consisted of a process modeling task as well as a questionnaire with items relating to flow, intrinsic motivation, extrinsic motivation, technology acceptance, and self-efficacy. Officially, the experiment was described as a "process modeling self-assessment", and the reason for its execution that was communicated to students was to review the success of the *Introduction to IS* lecture in teaching process modeling. Thus, no indication about its examination of the impacts of gamification was given.

The modeling task was adapted from the final exam of another lecture on process modeling and BPM which is part of the same curriculum at the University of Münster. As this lecture is typically attended by third-semester students, its difficulty level can be described as comparably high with regards to the expected proficiency level of the students after their first term. Overall, the business process to be modeled involved sequential activities, alternatives and parallelism with splits and corresponding synchronizations, an iteration, and references to roles and resources. In addition to representing the process correctly, students were also asked to ensure the high levels of readability, understandability, and completeness of their models. Originally, the task description was formulated in English and thus had to be translated into German to correspond with the language of the lecture. The source version of the modeling task reads as follows:

> *Since November 2010, citizens of Osnabrück have the possibility to obtain not only the conven-tional ID card, which exists with or without a signature component, but also an ID card with biometric safety features. To get an ID, the citizen has to come to the department "citizen service" at the administration office. If the citizen does not have all the required documents, he is sent back home; otherwise, the clerical assistant starts with data collection. To that extent, the assistant uses an electric cutting machine to automatically cut the photograph for the ID to the right format. At the same time, the assistant captures the personal data of the citizen in the soft-ware MESO, which requires the citizen to authenticate themselves with an existing piece of identification such as the old ID card, a passport, or a certificate of birth.*
>
> *Subsequently, the assistant enquires whether the citizen wishes his ID card to include the new biometric safety features. If this is desired, two fingerprints are taken using the software module MESO-FP. This module checks the quality of the fingerprints. If the prints are unreadable, the capturing process is repeated. Once the capture has succeeded, the assistant prints an overview of the personal data and glues the photograph onto the former. Lastly, the citizen is asked to pay the fee for the new ID card and the completed application data is forwarded to the German Federal printer.*

The experiment was conducted in a computer pool at the premises of the Department of Information Systems, and all materials required by students were prepared and configured ahead of time, thereby ensuring equal conditions for all participants. This most notably included a computer on which the required version of the Horus Business Modeler was already installed, a copy of the survey, and writing materials. Before the beginning of the experiment, the questionnaire was distributed to students in closed envelopes, and instructions were given only to withdraw individual pages as instructed. The distribution followed a deterministic pattern that ensured that half of the participants received a version of the questionnaire with server credentials that made gamification available, whereas the other half received account details for which gamification was disabled. By this mechanism, students were subdivided into the experimental and control group. After a brief introduction, students were informed that participation in the experiment was fully anonymous, and that no personal information would be collected. Subsequently, students were asked to extract and fill in the first page from the envelope, which contained items regarding basic demographic information. Next, the second page was withdrawn and the participants used the provided information to connect to the indicated server repository.

Once all students signaled that they had successfully logged in, a timer measuring 45 minutes was started, and students commenced working on the modeling task using the process description on the third page. After 15 minutes, students were instructed to pause modeling, withdraw the fourth page with question items relating to flow from the envelope, fill it in, and then immediately continue modeling. After the remaining 30 minutes had elapsed, all students saved their work results and exited the HBM. Afterwards, participants extracted and completed the remaining three pages with questions relating to intrinsic motivation, extrinsic motivation, technology acceptance, and self-efficacy from the envelope. All pages were reinserted into the envelope, which was then sealed and exchanged for the recompense of €20. While the task was ongoing, an ongoing attempt was made to uphold exam conditions and minimize any communication between students as far as possible.

8.3.2 Data and Measurement

The measurement instrument used for the experiment was constructed by combining previously-published and validated questionnaires for flow, intrinsic motivation, and technology acceptance, four self-developed items related to extrinsic motivation, a self-developed scale process modeling self-efficacy, and question items for basic demographic information. Questions were presented in a randomized fashion, thereby reducing the likelihood of any response patterns related to item ordering. The reliance on established measurement instruments means the overall questionnaire can reasonable be expected to exhibit high levels of reliability and validity. In the following, only a brief description of the questionnaire is provided; for the full instrument as it was distributed to students, the reader is referred to Section D.1 in the appendix.

Demographic information. At the beginning of the experiment, participants provided the following information about themselves: age, gender, average number of hours playing computer and video games per week, number of semesters of process modeling experience, self-reported process modeling expertise measured on a scale from 0 (beginner) over 50 (advanced) to 100 (experts), preparation for the experiment (yes or no), and, if applicable, details about any preparatory activities as a free text.

Flow. Due to its availability in German, flow was measured using the *Flow-Kurzskala (FKS)* presented by Rheinberg et al. [RVE03]. Alternative measurement instruments such as the Flow State Scale-2 (FSS-2) and the Dispositional Flow Scale-2 (DFS-2) [JE02] are used more widely and have also been applied to the context of gamification [HK14], but are only available in English. The FKS subdivides flow experiences into two components: *smooth progression* ("glatter Verlauf" in German), and *absorbedness* ("Absorbiertheit" in German). Both factors relate to the various characteristics of flow experiences as outlined in Section 3.5.3. Additional question items measure *anxiety*, which is expected to affect flow negatively, and the *skill-challenge balance*, i. e., whether individuals perceive their skills to be appropriate for the level of difficulty that a task presents them with. The latter is once again subdivided into three questions relating to the *difficulty* of the task, the perceived *skill level* of the individual with regard to the task, and the *balance* between these two components.

Intrinsic motivation. To assess the subjective experience of the participants while working on the task, the *intrinsic motivation inventory (IMI)*[6] measurement instrument provided by Deci and Ryan was used. Specifically, items for the constructs *interest/enjoyment*, *perceived competence*, *pressure/tension*, and *effort/importance* were included. The interest/enjoyment subscale is the most important component of the questionnaire and represents the self-reported degree of intrinsic motivation. Perceived competence is one of the three basic human needs according to Self-determination Theory and is expected to be a positive predictor of intrinsic motivation. Inversely, pressure/tension is theorized to affect intrinsic motivation negatively. Lastly, effort/importance is a secondary component of the IMI and is concerned with

[6] See: http://selfdeterminationtheory.org/intrinsic-motivation-inventory/. Last accessed: 2017-04-22.

the extent to which participants value a certain task. The IMI is offered in English, and was thus translated into German using the *back-translation* method [Bri86, Bri70]. Back-translation consists of multiple rounds of translating the text back and forth between the source and target languages by bilingual individuals. If the final version in the original language is very close to the source text, the last translation can be expected to be reasonably faithful; otherwise, inconsistencies must be resolved. Here, two rounds of back-translation were conducted (i. e., English-German-English-German) with the help of doctoral students at the Department of Information Systems, and the end result required only a few additional modifications.

Extrinsic motivation. As participation in the experiment was coupled to a monetary reward of €20, extrinsic motivation can be expected to be an influential factor in the decision of many students to partake. Therefore, question items for the two "lowest" levels of extrinsic motivation were included, namely *external regulation* (performing behavior solely for external reasons, such as rewards or commands) and *introjected regulation* (acting out of internal control, for instance to avoid guilt or to satisfy one's own pride) [RD00b].

Technology acceptance. Based on the most recent published version of the Technology Acceptance Model—the TAM3—items for *perceived usefulness* (belief that the HBM will help users to perform their job better), *perceived ease of use* (belief that the use of the HBM will be free of effort), *output quality* (belief that the HBM creates high-quality output), and *behavioral intention* (intentions regarding future use of the HBM) were included [Dav89, VB08]. Except for output quality, all of these constructs are also represented in the CUE model used in the field experiment, thereby enabling a comparison of results. The question items are offered in English, and were thus translated into German by the author.

Self-efficacy. Bandura states that specific scales of perceived self-efficacy should be constructed specifically for the domain to which the object under examination belongs [Ban06]. Thus, following the guidelines and recommendations specified in [Ban06], ten items for the measurement of *process modeling self-efficacy* were conceptualized. The respective items relate to the objective, present belief of respondents that they can accomplish tasks such as identifying activities in a textual process description, modeling sequences and alternatives, and creating a model of high quality. All items are measured on a scale from 0 to 100, with 0 denoting a lack of confidence, and 100 high confidence regarding a particular statement. Consequently, the overall perceived process-modeling self-efficacy of a subject can be calculated as the sum of all items, and may thus range from 0 to 1000.

A summary of the measurement instrument is provided in Table 8.23, with the columns containing the names of the constructs, the numbers of question items used to measure each construct, the codes of the respective items, and the scales on which items were measured. With a few exceptions, all items were measured on a 7-point Likert scale ranging from the complete rejection (1) to the complete agreement (7) with the given statements. In the context of flow, items related to the skill-challenge balance were measured on a 9-point Likert scale as defined in the FKS, and items related to process modeling self efficacy were measured from 0 to 100 as recommended by Bandura [Ban06]. Furthermore, the scores for most constructs were determined by averaging all relevant question items, with the exceptions once again being skill-challenge balance and self-efficacy. For the former, the individual items stand for themselves, and thus no overall construct score is computed. For the latter, the overall indicator of process modeling self-efficacy is determined as the sum of all item scores. More detailed information, including the exact wording of each measurement item, can be found in Section D.2 of the appendix.

In total, the experiment yielded 54 questionnaires, with 29 corresponding to the experimental group, and 25 to the control group as summarized in Table 8.24. Of these, nine (i. e., 16%) were incomplete by missing responses for one or more items. No pattern could be detected for the missing values, and thus the dataset was considered appropriate for the restoration of missing values through multiple imputation [Rub87]. Using this method in SPSS, missing values were estimated from the remaining data, and thus five alternative complete samples were created. As aggregating the results of these imputations is generally discouraged, analyses were conducted for each set independently, and any differences are reported as required. To detect careless responses, the Mahalanobis' distance measure was computed for all questionnaires [Mah36], thereby allowing for the exclusion of one multivariate outlier from further consideration. After these steps, 53 complete samples were obtained for each imputation and used as input for the following analysis phase. The impacts of this sample size on the interpretation of results is discussed in Section 8.4.2.

Table 8.23: Laboratory experiment: Survey overview.

Construct	Items	Codes	Scale
Flow			
Smooth progression	6	FG1-FG7	7-point Likert
Absorbedness	4	FA1-FA4	7-point Likert
Anxiety	3	FB1-FB3	7-point Likert
Skill-challenge balance	3	FP1-FP3	9-point Likert
Intrinsic Motivation			
Interest/enjoyment	7	II1-II7	7-point Likert
Perceived competence	5	IC1-IC5	7-point Likert
Pressure/tension	5	IP1-IP5	7-point Likert
Effort/importance	5	IE1-IE5	7-point Likert
Extrinsic Motivation			
External regulation	2	ER1-ER2	7-point Likert
Introjected regulation	2	IR1-IR2	7-point Likert
Technology Acceptance			
Perceived usefulness	4	PU1-PU4	7-point Likert
Perceived ease of use	4	PEOU1-PEOU4	7-point Likert
Output quality	3	OUT1-OUT3	7-point Likert
Behavioral intention	3	BI1-BI3	7-point Likert
Self-efficacy			
Process modeling self-efficacy	10	PMSE1-PMSE10	Integer 0-100

Table 8.24: Laboratory experiment: Set of participants.

	Responses				
Month	*Total*	*Excluded*	*Included*	**Gamification**	**Included**
2016-03	54	1	53	✔	28
				✖	25

8.3.3 Validity and Reliability

Before conducting the data analysis, the reliability and validity of the sample were examined to assess the consistency and accuracy of the measurements. The results of these tests are presented in the following two subsections.

Reliability

The internal consistency of the measurement instrument was analyzed following [Fie09] by computing Cronbach's α and inter-item correlation matrices for all constructs. The results of these examinations are as follows.

Flow. For the construct *smooth progression*, Cronbach's α was calculated as 0.809 for the original dataset, and 0.799 or higher for all imputations. However, the question items FG2 and F3 exhibited low correlations with all remaining items, and were thus deleted from the subscale. This yielded an increase of Cronbach's α to 0.822 or higher for all version of the dataset. Furthermore, all inter-item correlations of the remaining questions fall into an acceptable range (0.428-0.645), and thus no further adjustments were made. For the construct *absorbedness*, Cronbach's α was calculated as 0.330 for the original dataset, and 0.274 or higher for all imputations. This is, by all standards, an unacceptable result [Pet94], and no single item could be identified as responsible for this level of unreliability. Similarly, the reliability of the construct *anxiety* did not exceed a Cronbach's alpha of 0.575 for any version of the dataset with no possibility for improvement through item deletion. An EFA was conducted to examine alternative possibilities to factorize the question items, but no meaningful model could be determined. Therefore, both of these subscales were dropped from the subsequent analysis. Possible reasons for the apparent inadequacy of the measurement instrument to measure these two constructs are examined in Section 8.4. Lastly, as the question items relating to *skill-challenge balance* are intended to be used individually, an examination of their reliability was not necessary.

Intrinsic motivation. For the construct *interest/enjoyment*, Cronbach's α was calculated as 0.833 or higher for all versions of the dataset. Furthermore, all correlations fell into an acceptable range, and thus no adjustments were made. For the construct *perceived competence*, Cronbach's α was calculated as 0.758 or higher for all versions of the dataset. However, the correlation of item IC5 was below 0.3 for two items, and thus it was deleted from the subscale [Fie09]. This resulted in an increase of alpha to 0.800 for the original dataset, and 0.778 or higher for all imputations. Furthermore, all inter-item correlations were improved to an acceptable range. For the construct *pressure/tension*, Cronbach's α was calculated as 0.835 or higher for all versions of the dataset. While the inter-item correlation of IP1 and IP3 fell below the level of 0.3, all remaining correlations were acceptable, and thus no item was deleted. For the construct *effort/importance*, Cronbach's alpha was calculated as 0.532 for the original dataset and 0.527 or higher for all imputations. After the removal of the items IE2 and IE5 with the lowest inter-item correlations, the value of alpha could be improved to 0.655 or higher for all versions of the dataset, and all remaining correlations were greater than 0.3. It should be noted that this still does not indicate satisfactory levels of consistency according to the most common thresholds for Cronbach's α [Pet94].

Extrinsic motivation. For the construct *extrinsic regulation*, Cronbach's α was calculated as 0.814 for the original dataset, and 0.782 or higher for all imputations. Furthermore, the correlation between both items was just below 0.7, and thus no measures were required. For the construct *introjected regulation*, Cronbach's α was calculated as 0.701 for the original dataset, and 0.696 or higher for all imputations. Furthermore, the correlation between both items was computed as 0.540, and thus no measures were required.

Technology acceptance. For the construct *perceived usefulness*, Cronbach's α was calculated as 0.899 for the original dataset, and 0.897 or higher for all imputations. Furthermore, all correlations fell into an acceptable range, and thus no adjustments were made. For the construct *perceived ease of use*, Cronbach's α was calculated as 0.734 for the original dataset, and 0.733 or higher for all imputations. The item PEOU2 exhibited two inter-item correlations lower than 0.2, and was thus removed. After this adjustment, Cronbach's alpha was determined as 0.791 for all versions of the dataset. For the construct *output quality*, Cronbach's α was calculated as 0.758 for the original dataset, and 0.724 or higher for all imputations. Furthermore, all correlations fell into an acceptable range, and thus no adjustments were made. For the construct *behavioral intention*, Cronbach's α was calculated as 0.888 for all versions of the dataset. Furthermore, all correlations fell into an acceptable range, and thus no adjustments were made.

Self-efficacy. For the construct *process modeling self-efficacy*, Cronbach's α was calculated as 0.895 for the original dataset, and 0.891 or higher for all imputations. However, this does not necessarily indicate a high level of consistency, as alpha is known to be positively influenced by a larger number of question items—in this case 10 [TD11]. An examination of the inter-item correlation matrix revealed that half of the items were well-correlated with each other, but not with the remaining half. Therefore, an EFA was conducted to check for the existence of multiple latent variables. Principal component analysis was used as an extraction method, varimax and oblimin as rotation methods, and the Kaiser criterion for component selection. Both the Kaiser-Meyer-Olkin test as well as Bartlett's test of sphericity indicate the appropriateness of the sample for factor analysis according to the guidelines summarized by Field [Fie09]. The analysis yielded two factors accounting for 68% of the variance of the underlying data, with all factor loadings being larger than 0.6 except for the item PMSE1, which loads onto one factor with a value of 0.43:

- *Model extraction*: The survey items PMSE1 and PMSE3-6 are related to expectancies of being able to extract a process model (including activities, objects, and control flow) from a textual process description. For this construct, Cronbach's α was calculated as 0.848 or higher for all versions of the dataset. Furthermore, all inter-item correlations with one exception with a value of 0.293 fell into an acceptable range, and thus no items were deleted.

- *Quality assurance*: The survey items PMSE2 and PMSE7-10 are related to expectancies of being able to create a process model of high quality. For this construct, Cronbach's α was calculated as 0.885 for all versions of the dataset. Some inter-item correlations exhibited high values, which can be attributed to the partial redundancy of some questions. However, no further modifications were made.

After these modifications, the internal consistency of the measurement instrument can be described as satisfactory, with many coefficients exceeding the stricter threshold of 0.8, and only Cronbach's alpha for the construct *effort/importance* falling shy of the common cutoff value of 0.7. Overall, this is interpreted as a sufficient reliability of the utilized questionnaire. The result of

Table 8.25: Laboratory experiment: Survey after reliability analysis.

Construct	Items	Code	Construct	Items	Code
Flow			**Technology Acceptance**		
Smooth progression	4	FG	*Perceived usefulness*	4	PU
Difficulty level	1	FD	*Perceived ease of use*	3	PEOU
Skill level	1	FS	*Output quality*	3	OUT
Skill-challenge balance	1	FB	*Behavioral intention*	3	BI1
Intrinsic Motivation			**Process Modeling Self-efficacy**		
Interest/enjoyment	7	II	*Model extraction*	5	SEME
Perceived competence	4	IC	*Quality assurance*	5	SEQA
Pressure/tension	5	IP			
Effort/importance	3	IE			
Extrinsic Motivation					
External regulation	2	ER			
Introjected regulation	2	IR			

the reliability analysis is summarized in Table 8.25. Notable deviations from the original questionnaire as shown in Table 8.25 include the removal of individual question items, the rejection of two constructs related to flow, the division of process modeling self-efficacy into two related factors, and the assignment of new construct codes where required. Furthermore, the three question items concerned with the balance of skill and difficulty level are listed separately.

Validity

Analogously to the measurement instrument employed for the field experiment, *face validity* and *content validity* were not assessed separately, but are assumed to be satisfied due to the predominant use of question items adapted from previously-published, validated questionnaires. Furthermore, *construct validity* was examined by testing for *factorial validity*. To that extent, a path model for conducting a CFA for the central part of the questionnaire was created with the software IBM SPSS Amos. This model includes the 10 constructs and 36 items that were not removed after the examination of reliability, with the exception of flow and self-efficacy as isolated constructs measured on separate pages. The results of the analysis reveal an unsatisfactory fit of the model

Table 8.26: Laboratory experiment: Model fit of the CFA.

Fit index	Recommended	Model	
χ^2 per degree of freedom	<3	1.612	✔
Goodness of fit index	>0.9	0.577	✖
Adjusted goodness of fit index	>0.8	0.492	✖
Comparative fit index	>0.9	0.737	✖
Normed fit index	>0.9	0.535	✖
Root mean square error of approximation	<0.5	0.47	✔

with the dataset as evidenced by a violation of most model fit indices provided by Amos when compared to "rule of thumb" values for IS research [GSB00] (see Table 8.26). The main reason for this can be found in the sample size of 53, which is too small to estimate all distinct parameters of the model. More concretely, Kline summarizes the following minimum sample size guidelines from relevant literature [Kli11]: the subjects-to-variable ratio should be as high as possible, with common ratios being 20, 10, and 5; the number of cases should not be smaller than 200; sample sizes smaller than 100 may only suffice for very simple path models. As the dataset that was obtained from the experiment misses these thresholds by a large margin, the unsatisfactory model fit is to be expected [Tan87], and thus factorial validity cannot be demonstrated.

Following [PLX07] and [KH14], *convergent validity* is also indicated if the Cronbach's α satisfies the recommended threshold for all constructs. This is the case for the measurement instrument at hand, the only exception being the factor *effort/importance* with an alpha value of 0.655. As an alternative for assessing *discriminant validity*, the Pearson correlations of all items in the survey can be examined, although this procedure is highly subjective in nature [Kli11]. The respective values are depicted in Figure 8.2, with individual cells representing correlations between question items. Furthermore, black borders enclose survey items that measure the same construct, and any items on the upper diagonal outside these blocks are deemphasized, as they mirror the values in the bottom-left half of the matrix. In conformance with Cohen's recommendations, correlations with a value of at least 0.5 are considered high and highlighted in orange, whereas correlations of at least 0.3 are considered moderate and highlighted in green [Coh92]. Additionally, correlations of at

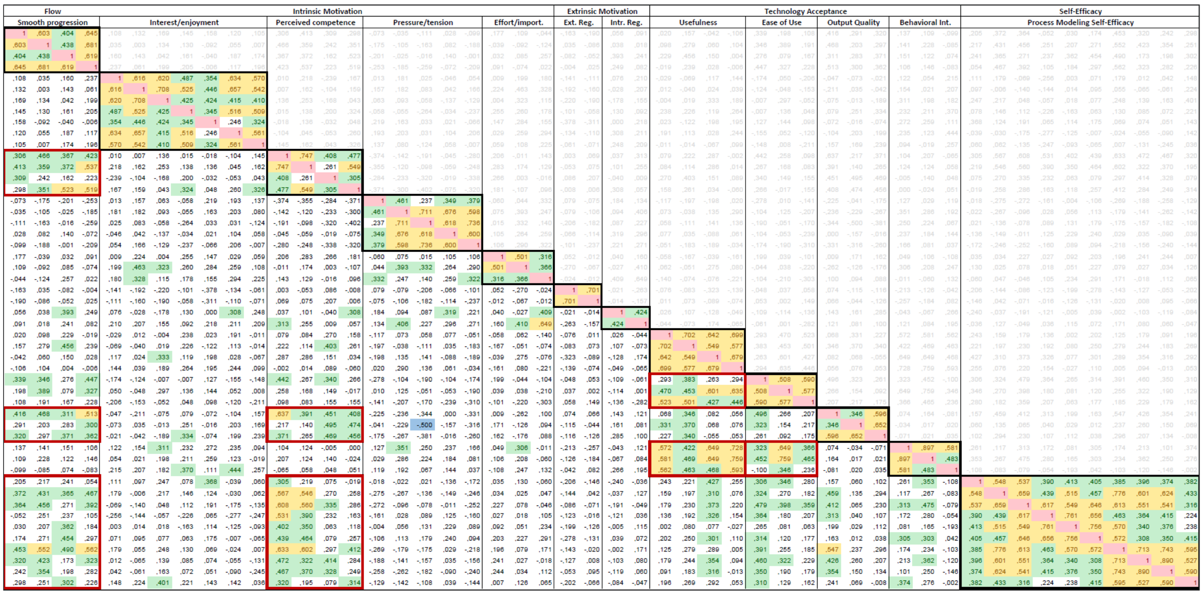


Figure 8.2: Laboratory experiment: Inter-item correlation matrix.

most -0.5 are displayed with a blue background color. Lastly, red values on the diagonal depict the correlations of individual variables with themselves.

In this matrix, most occurrences of high correlations can be found along the diagonal in blocks of questions corresponding to the same factor. Furthermore, most items are more strongly correlated with items of the same construct than with items of other constructs, i. e., the largest correlations for most rows and columns can be found within black borders. Notable exceptions are highlighted by dark-red rectangles, and can be explained by the relationships between the respective factors. For instance, moderate-to-high correlations can also be found at the intersections of items belonging to the constructs *smooth progression*, *perceived competence*, *output quality*, and *process modeling self-efficacy*. This can be attributed to the fact that all four concepts are related to the performance of students working on the modeling task. A similar pattern exists for the constructs *perceived usefulness*, *perceived ease of use*, and *behavioral intention* which are adopted from the Technology Acceptance Model 3 (TAM3). This conforms to the conceptualization of this model, which posits that ease of use is a predictor for usefulness, and both individually explain the intentions of the users of a system about its future use [VB08]. Overall, the results of the reliability analysis and the patterns in the inter-item correlation matrix are interpreted as indications for both convergent and discriminant validity.

8.3.4 Results

Initial insights into the impacts of gamification on flow, motivation, technology acceptance, and self-efficacy can be gained from the descriptive statistics presented in Table 8.27. Here, the means, standard deviations (SD), and medians of the construct scores collected through the measurement instrument are depicted. Additionally, the final column indicates the relative increase or decrease of the mean construct score in the presence of gamification. For instance, a difference of 0.08 indicates that the aggregated value of the respective factor is 8% higher than its non-gamified counterpart. For most constructs, possible values range from a minimum of 1 to a maximum of 7, with 4 representing a neutral "neither ... nor" response. Exceptions are the final three items related to flow, which are measured on a scale from 1 to 9, and both items related to self-efficacy, which are computed as the sums of five items with values ranging from 0 to 100, any may thus exhibit scores of 0 to 500. The "best" value of most constructs can be found at the higher end of the scale, the exceptions being *difficulty* and *pressure/tension*, where lower values are better, and *balance*, where the middle value of 5 is optimal. Overall, the following observations about the data can be made:

- **Overall**. With a few exceptions, most mean and median construct values fall within an interval of ± 1 around the neutral, middle value in the questionnaire. However, most standard deviations are above 1.0, which indicates that students did not strongly agree in their responses.
- **Flow**. Evidence for flow experiences in the data is weak, and differences between both groups are negligible. Looking at the preconditions for flow, it can be seen that students rated the difficulty of the task higher than their respective skill levels. Furthermore, both groups indicated that they found the task to be slightly too difficult, which may be one indicator for the absence of flow. In comparing both groups, it can be seen that those students who worked with gamification exhibit higher values for both difficulty and skill. This may be a result of the provided real-time

Table 8.27: Laboratory experiment: Descriptive statistics.

	Gamification, $N = 28$			~~**Gamification**~~, $N = 25$			
Construct	**Mean**	**SD**	**Med.**	**Mean**	**SD**	**Med.**	**Diff.**
Flow							
Smooth progression	4.32	0.93	4.50	4.33	1.21	4.50	=
Difficulty	4.61	1.83	5.00	4.25	1.65	4.00	0.08△
Skill	4.29	1.70	4.00	3.92	1.81	4.00	0.09▲
Balance	5.43	1.07	5.00	5.58	1.10	5.00	-0.03▼
Intrinsic Motivation							
Interest/enjoyment	4.61	0.95	4.71	4.46	1.09	4.79	0.03▲
Perceived competence	3.40	1.09	3.50	3.58	1.14	3.75	-0.05▽
Pressure/tension	3.65	1.20	3.60	3.20	1.40	3.30	0.12△
Effort/importance	4.36	1.08	4.17	4.25	1.11	4.33	0.03▲
Extrinsic Motivation							
External regulation	4.88	1.49	5.00	4.83	1.81	5.00	0.01△
Introjected regulation	2.77	1.40	2.50	2.54	1.23	3.00	0.08▲
Technology Acceptance							
Perceived usefulness	4.99	1.03	5.00	4.49	1.22	5.00	0.10▲
Perceived ease of use	4.58	0.67	4.50	4.13	1.51	4.33	0.10▲
Output quality	3.08	1.11	3.00	3.01	1.10	3.00	0.02▲
Behavioral intention	4.91	1.08	5.00	3.83	1.62	3.33	0.22▲
Self-efficacy							
Model extraction	308	90	310	309	99	309	=
Quality assurance	302	77	300	307	112	325	-0.02▽

quality feedback, which reveals the quality defects that models have, but also gives modelers the necessary tools for model improvement.

- **Intrinsic motivation**. Students who participated in the experiment reported moderate levels of intrinsic motivation (indicated by the construct *interest/enjoyment*) independently of the group to which they belonged. However, their perceived competence for the task is comparably low, which is reflected in the constructs of flow measuring the balance of difficulty and skill. Furthermore, this observation could explain why students did not experience stronger levels of intrinsic motivation. The data also indicates that the participants did not feel pressured or anxious while working on the task, which may have contributed positively towards intrinsic motivation. Interestingly, pressure and tension are higher for students who have worked with gamification, which may once again be a consequence of the real-time quality feedback provided by the gamification module. Lastly, both groups invested "normal" levels of effort into the task, i. e., they did not consider it particularly important or unimportant.

- **Extrinsic motivation**. The level of external regulation indicated by both groups exceeds their respective intrinsic motivation by a small margin. Consequently, it can be argued that students participated in the experiment mostly for the monetary reward, but were almost equally interested in the task itself. However, the reported levels of introjected regulation are comparably small, and thus it cannot be concluded that the students participated to satisfy their own pride or to demonstrate their capabilities. However, it should be noted that introjected regulation was higher for those students having worked with gamification.

- **Technology acceptance**. The strongest impacts of gamification can be observed for constructs related to technology acceptance. Specifically, both the perceived usefulness as well as the perceived ease of use of the HBM was rated 10% higher in the presence of game design elements. Furthermore, both constructs exhibit moderate values slightly above the neutral response. This is not the case for output quality, which exhibits values coinciding with the skill subscale of flow and the perceived competence subscale of intrinsic motivation. Thus, students did not believe that the models they created had high levels of quality. Lastly, the average intention of students to continue using the HBM is 22% higher in the presence of gamification, which may indicate the existence of a direct relationship between both that is independent of any other measured construct.

- **Self-efficacy**. The reported levels of process modeling self-efficacy are almost constant at a moderate level (300 out of a maximum value of 500) for both groups of participants and both measured subscales. Consequently, it can be argued that the difficulty of the task and the low perceived skill levels and competence of the participating students diminishes any positive impacts that gamification could have on self-efficacy. Inversely, self-efficacy has also been described as a predictor for the flow state and the effort that individuals exert to accomplish a certain task (see Section 3.5.5), which may explain the lack of an impact of gamification on related constructs.

An examination of the dataset for normality using the Kolmogorov-Smirnov and Shapiro-Wilk tests reveals that most constructs follow a normal distribution for both groups with an alpha level of 0.05. However, some exceptions exist, most notably the variables FB (both groups), II (both groups), IP (no gamification), ER (no gamification), and IR (gamification). A visual examination of the data indicates that the problematic constructs are borderline cases, and thus an additional test for normality was conducted following [Fie09] based on the skewness and kurtosis of their distributions. From the results of this test, it can be concluded with an alpha level of 0.05 that only FP3 (no gamification) and II (gamification) exhibit a significant difference from normality. Thus, an *independent t-test* is more appropriate for the data at hand, and was thus conducted instead of the *Mann-Whitney-U* test utilized in Section 8.2.2 to check for the existence of statistically significant differences between the means of both subsamples. Besides normality, the assumptions of this test are homoskedasticity (i. e., homogeneity of variances) and stochastic independence of the data from the experimental and control groups [Fie09]. However, SPSS can produce an adjusted result when the distributions are heteroskedastic, which is reported for constructs where this is the case.

The results of this test are summarized in Table 8.28 according to the scheme presented in Section 8.1 with the following alterations: instead of the U-value and Z-value of the Mann-Whitney-U test, the t-statistic and the degrees of freedom (df) are reported. As for the other hypothesis tests, the utilized effect size measure is Pearson's correlation coefficient r; however, following [Fie09], it is calculated as $r = \sqrt{t^2/(t^2 + df)}$. The results presented in Table 8.28 illustrate that neither of the differences in means for any construct related to flow, intrinsic motivation, extrinsic motivation, and self-efficacy is significant. Therefore, it cannot be concluded that the observed variations are the result of the experimental condition (i. e., the utilization of gamification; also called the *treatment* [Cre09]) rather than chance. The possible underlying reasons for this are examined in more detail in Section 8.4. Looking at factors related to technology acceptance, statistically significant differences can be observed for all factors except output quality, i. e., students who worked on the task using the gamified version of the HBM perceive its usefulness and ease of use as higher (medium effect size), and express a stronger behavioral intention to continue using the tool in the future (large effect size). Interestingly, this mirrors the results of the field experiment, where no impact of gamification on usefulness and behavioral intention was detected, and a medium effect of gamification on usability (i. e., ease of use). This may either be the result of different conditions in both experiments, or an indication that the constructs measured by both questionnaires are distinctly different despite their identical names. The latter is not uncommon, and the belief that two things should be the same because they share the same name is also known as a *jingle fallacy* [PP91]. While the results presented in Table 8.28 provide the figures obtained for the original dataset with missing values, the magnitudes of all significant effects are identical for all imputations.

Table 8.28: Laboratory experiment: Hypothesis test results.

Construct	Hom.	t	df	p-Value	Sig.	Effect	Size
Flow							
Smooth progression	✔	-0.120	51	0.452	-	-	-
Difficulty	✔	-0.734	50	0.233	-	-	-
Skill	✔	-0.680	51	0.250	-	-	-
Balance	✔	0.444	51	0.329	-	-	-
Intrinsic Motivation							
Interest/enjoyment	✔	-0.622	51	0.268	-	-	-
Perceived competence	✔	0.279	51	0.391	-	-	-
Pressure/tension	✔	-1.070	51	0.145	-	-	-
Effort/importance	✔	-0.213	51	0.416	-	-	-
Extrinsic Motivation							
External regulation	✔	0.011	51	0.496	-	-	-
Introjected regulation	✔	-0.788	51	0.217	-	-	-
Technology Acceptance							
Perceived usefulness	✔	-1.882	51	0.033	**	0.2549	M
Perceived ease of use	✖	-1.634	32	0.056	*	0.2774	M
Output quality	✔	-0.368	51	0.357	-	-	-
Behavioral intention	✖	-3.000	40	0.002	***	0.4281	L
Self-efficacy							
Model extraction	✔	0.048	51	0.481	-	-	-
Quality assurance	✖	-0.066	41	0.474	-	-	-

8.4 Discussion

In this section, the results of both experiments are discussed. To that extent, Section 8.4.1 summarizes their key findings against the backdrop of the challenges and goals of Social BPM outlined in Chapter 4. Section 8.4.2 and Section 8.4.3 then outline the limitations of the experiments, and discuss possible threats to their internal and external validity.

8.4.1 Summary of Results

The gamification module was designed to address certain challenges that occur in the context of Social BPM, namely ensuring satisfactory levels of model quality, motivating potential contributors to perform activities relevant to Social BPM, and providing appropriate measures for training and education. In this section, the findings gained from both experiments that were carried out are summarized and mapped against these three problems.

Ensuring Model Quality

Due to the instrumentation of the principles of social software, process modelers in "proper" manifestations of Social BPM may not only be BPM experts, but also novice users without prior modeling experience. This presents the following two problems. Firstly, maintaining high levels of model quality has been recognized as a difficult task when inexperienced individuals are involved [Ros06a, EGH+10, MRvdA10, FL15]. Secondly, model quality—in particular understandability—becomes even more important in such settings, as the skill set BPM novices can refer to while interpreting process models is highly limited [BRvU00]. The possible contribution that gamification can make to this challenge was derived from the fact that process modeling exhibits characteristics that are similar to a game, and may thus have the same, performance-maximizing function if structured as a game-like activity. To that extent, the overarching goal of creating high-quality process models was subdivided into smaller tasks corresponding to individual quality metrics, and appropriate real-time quality feedback was provided.

The impacts of gamification on model quality were examined through the field experiment described in Section 8.2.4. By analyzing the descriptive statistics (i. e., medians and means) presented in Table 8.11–tbl:evaluation-field-experiment-models-median-2, initial insights into the effect of gamification on quality metrics on an aggregated level could be gained. However, the most notable findings were derived from the three hypothesis tests that were carried out, which compared data from 2015 (with and without gamification), 2016 (with and without gamification), and 2015 and 2016 (experimental group for both). The core results revealed by these tests are the following:

- *Gamification has a limited impact on trivial metrics.* When creating a process model, modelers automatically try to avoid readability problems such as edge crossings, excessive use of edge bends, and node overlaps. This is reflected in the high average values of the respective metrics, which are close to 1.0 across all years independently of gamification. Therefore, it can be concluded that a subset of "trivial" metrics exists for which gamification does not have a significant impact. However, it should be noted that most models created in the context of the field experiment were rather small, and for larger modeling projects, different outcomes might be observed.

- *Gamification can help modelers to work with complex metrics.* On the opposite end, some quality metrics encapsulate more complex ideas, and may thus be more difficult to comprehend, especially for novice modelers. For instance, the metrics connector mismatch, *control flow complexity*, and *token split* are best understood with foundational knowledge about the concepts that they embody. However, results of 2016 demonstrate that novice modelers may still operationalize real-time quality feedback together with specific goals and rudimentary help features to improve the values of metrics that they don't necessarily fully understand.

- *Gamification reminds modelers to create complete process models.* The strongest impacts of gamification could be observed for quality metrics related to model completeness. In fact, statistically significant effects for individual metrics could only observed with regard to completeness in 2015, and exhibited the largest effect sizes in 2016. This indicates that model completeness as a quality characteristic is easy for novice modelers to operationalize, and thus motivates them to set and pursue high quality goals. A possible explanation is offered by self-efficacy theory, which suggests that individuals set personal aims for themselves that coincide with their perceived skill level [LL06]. For students, model completeness may present an optimal goal, as it is less trivial than many readability metrics, yet not as complicated as some complexity metrics.

- *The impacts of gamification are often small.* As the effect sizes in Table 8.15 and Table 8.16 illustrate, the magnitudes of the effects caused by gamification are often comparably small, even when utilizing more generous bounds suggested by recent research [GS16]. Whether this is a limitation of the current implementation of the gamification module, of the impacts of gamification in the context of the field experiment, or of gamification as a tool for process modeling overall cannot be decided without further information and is thus a possibility for future research.

- *Gamification requires specific goals.* A comparison of the numbers presented in Table 8.17 Table 8.18 reveals strong differences in the effects of gamification between 2015 and 2016. These variations are statistically significant for all quality metrics besides edge bends and node occlusion, and predominantly exhibit medium or large effect sizes. As pointed out in Section 8.2.1, one of the most significant differences between these two iterations of the lecture lies in the extent and specificity of the requirements for submissions that were communicated to students. Therefore, the diverging results can be explained by goal-setting theory, which posits that more specific goals have a higher impact on performance than goals that are rather vague or abstract, such as the directive to simply “do one’s best” [LL06].

- *Gamification can affect quality metrics without feedback.* As expected, the largest impacts of gamification could be observed for quality metrics for which students actually received quality feedback. However, as Table 8.16 shows, significant effects also exist for “hidden” quality metrics, most notably those related to model complexity. This is not surprising, as the quality concerns these metrics embody are not mutually exclusive, and thus improving one metric may change the value of others in either direction. For instance, the number of edge crossings in a process graph can be reduced by using edge bends to route arcs around problematic areas. Overall, this indicates that it is not necessary to overburden modelers with feedback for all metrics to achieve high levels of model quality. Instead, the interrelationships between these metrics should be studied in more detail to identify those with the highest importance.

In summary, it can be concluded that gamification is an effective tool for achieving higher levels of model quality even in the presence of unexperienced process modelers. However, the magnitude of its impacts strongly depends on *how* it is used. For example, comparing the results from 2015 and 2016 revealed that students exert more effort to improve model quality when provided with specific goals. Thus, for the design and implementation of a gamified solution, it should be kept in mind that game design elements must properly aligned with the context into which they are integrated to unfold their full potential.

Motivating Participation

One of the preconditions for successful Social BPM is the active involvement of all process stakeholders who can make relevant and meaningful contributions to business process models. Achieving this can be challenging, as the involvement in Social BPM represents a time-consuming task lying outside the core work activities of most employees, yet participation should optimally occur on a voluntary basis rather than being forced [EGH+10]. In this context, gamification is hypothesized to work as an additional layer of motivation that can help with making individuals interested in the BPM activities themselves (i. e., facilitate intrinsic motivation), or to provide them with extrinsic motivators such as tangible rewards, monetary gratification, or other positive (or negative) consequences based on some performance indicators. Overall, the impacts of gamification should therefore be reflected in quantitative indicators for the actual use of the HBM, certain motivational constructs that are theorized to be influenced by gamification, and the assessment of the HBM by its users.

Impact on actual system use. The impact of gamification on system use was examined through the field experiment described in Section 8.2.5. One of the key findings that was derived is that students who worked with the gamified version of the HBM conducted more modeling sessions (i. e., they logged into the server repository more often), interacted more with models

(i. e., by opening, closing, and saving them) and spent more time using the tool. Thus, there are sufficient grounds to state that gamification has resulted in higher levels of engagement and effort in the case study, as confirmed by the overall higher model quality achieved by gamification users. The second key finding relates to users exploring the different functions the HBM offers. Specifically, results reveal that students working with the gamified version of the tool fired a higher number of *different* types of events on average. This is equivalent to the statement that they used a higher share of the functionality that is tracked by the gamification module. One reasonable explanation for this is that the respective users performed actions that are preconditions for unlocking certain badges, such as providing basic profile information. Inversely, users of the non-gamified tool version saw little reason to do so, and thus no respective events have been recorded for them. Overall, it can thus be concluded that gamification can lead to a higher intensity and diversity of the way in which users utilize a software program.

Impact on motivational constructs. The impact of gamification on various indicators related to motivation was examined through the laboratory experiment described in Section 8.3.4. Contrary to expectations, results were not able to show a significant impact of gamification on flow, intrinsic motivation, extrinsic motivation, or perceived process modeling self-efficacy, which may be explained as follows. Firstly, the occurrence of flow experiences depends on a proper balance between the difficulty level of a task and the skills of the individual undertaking it [CAN05]. As the values of several constructs (e. g., difficulty, skill, balance, perceived competence, and self-efficacy) show, students may have perceived the task as too difficult, which constitutes a violation of the second condition, and may thus prevent flow. This perception was not influenced by the presence of gamification, which could be a result of the fact that inexperienced modelers focus more on the defects of their models that are revealed by real-time quality feedback rather than on what they have already accomplished. Secondly, the lack of an effect on intrinsic motivation indicates that gamification did not change how the modeling experiment satisfied the students' needs for competence and autonomy [RD00b]. In the case of competence, this may again be connected to the high perceived difficulty level of the task. Furthermore, the remuneration offered to students as a reward for their participation may have had a negative effect on autonomy, and thus intrinsic motivation. Thirdly, results for extrinsic motivation indicate that the primary motive for students of both groups to participate in the experiment was the monetary reward, and not a demonstration of their own skills. Lastly, the lack of an impact of gamification on perceived self-efficacy may once again be a result of the difficulty of the task, as experiences of mastery are one of the drivers for higher efficacy expectations [Ban77]. Clearly, such experiences are not possible if students doubt that they possess the necessary skills to accomplish a certain task. In summary, these individual results point to the importance of an appropriate difficulty curve as a precondition for positive motivational effects through gamification. For instance, an application such as Duolingo would arguably be far less motivating if it forced its users to learn the most difficult aspects of a language before its basics.

Impact on tool assessment. The impact of gamification on how students assessed the HBM was examined through both the field experiment and the laboratory experiment. In both cases, different measurement instruments were employed, namely the meCUE questionnaire for the former, and a subset of the TAM3 for the latter. The aggregated results of these surveys indicate that gamification can lead to higher perceptions of usability, ease of use, and visual aesthetics, heighten positive emotional responses while using a tool, serve as a cushion for possible frustrations, and vastly increase the intentions of users to continue utilizing a software in the future. It should be noted that the results of both measurement instruments differ; for instance, a positive effect of gamification on usefulness and behavioral intention was only determined in the laboratory experiment, but not in the modeling case study. This indicates that there may be additional factors not observed in the available data that mediate the impact of gamification. For example, whereas in the field experiment technical problems sometimes occurred (e. g., occasional server crashes spread throughout the semester), this was not the case in the laboratory experiment, which lasted less than an hour. Clearly, this may have affected how students assessed the HBM in these different contexts.

Educating and Training Participants

In its current manifestation, the implementation of the gamification module described in Section 7.4.1 realizes 57% of the elements that are covered by the "training lens" on the gamification design shown in Figure 6.18. However, most of the implemented elements only serve as preconditions for the most important features—tutorial tasks, quiz tasks, and the skill tree—which have not been realized yet. Therefore, data related to these elements could not be collected in either of the experiments, and thus an examination of the goals that were defined with regard to the education and training of individuals participating in Social BPM is considered future work.

8.4.2 Limitations

To properly interpret the findings gained from the two experiments, it is necessary to be aware of any limitations that the conducted research is subject to. The most important points that affect the results presented in this evaluation are the following.

Gamification as a bundle. It is reasonable to assume that not all components of the implemented gamification concept have influenced the measured constructs in the same way. For instance, the laboratory experiment consisted of a single process modeling task, and thus users have spent most of their time engaged with the real-time quality feedback. In contrast, neither the leaderboard nor the user profile has played any role for this task. This is slightly different for the field experiment, which was carried out over a much longer period of time and thus gave students ample opportunity to examine the entire variety of the features the gamification module offers. These discrepancies may in fact explain why the results of both studies differ in some aspects, for instance regarding usefulness and behavioral intention. Nevertheless, the implementation currently offers no possibility to isolate the effects of individual game design elements to judge their relative importance. This opens up a promising avenue for future research, as determining which elements work best in which contexts and for which users is an open problem [SF15b]. Possible inspiration to that extent can be drawn from various gamification studies (e. g., [MK14, Ham17, CF14a, LBC15, MBOT13a]) which have focused on the impacts of individual game design elements rather than bundles.

Implementation-specificity. In the previous parts of this section, any significant findings were attributed to "gamification" as a binary property that is either absent or present. However, the design of a gamified solution is a highly creative process [MWHA17], and thus for any given problem, an infinite number of possible designs may be created, each of which may again result in an infinite number of different implementations. Therefore, it would be more appropriate to state that certain effects could be observed for the concrete IT artifact that was created rather than for gamification as a concept. Consequently, it is possible that different implementations of the same design—or altogether different designs—would have led to other outcomes with better or worse performance. However, this limitation applies to all kinds of design-oriented research, and thus does not pose a significant problem.

Limitations of the measurement instruments. To explain the effects of gamification, researchers often refer to theories of human motivation such as those described in Section 3.5. Therefore, it has been recommended that applied gamification research should also evaluate the effectiveness of newly-created artifacts through measurement instruments that are adapted from those designed and validated in the respective disciplines, such as the intrinsic motivation inventory [SF15b]. Consequently, the use of the meCUE questionnaire in the field experiment can be considered suboptimal, and was mostly maintained as this questionnaire had already been in place for the lecture evaluation in previous years. While the instrument used for the laboratory experiment was based on already-established question items as recommended, some quality issues could be observed here as well. Firstly, the FKS constructs used to measure flow exhibited a low reliability, and thus some of them had to be discarded. This indicates that the respective survey, although designed to be universally applicable [RVE03], was not appropriate for the gamification context. Therefore, it may be necessary to adapt the respective instruments before they can be used without the need for concern. For instance, Hamari and Koivisto have proposed an alternative factorial structure for the DFS-2 instrument used to measure flow, which is more closely aligned with the characteristics of gamification [HK14]. Furthermore, it should be noted that question items related to intrinsic motivation and technology acceptance were translated from English to German. While great care was applied to create a translation that is as faithful as possible (for instance through the use of the back-translation method), the meaning and understandability of individual items may still have been altered.

Methodological limitations. Finally, some limitations also arise out of the concrete fashion in which this research project, and in particular the evaluation, was conducted. First, the sample sizes of the surveys carried out in the field experiment ($N = 79$) and the laboratory experiment ($N = 53$) are problematic. Specifically, the number of responses is below certain thresholds that must be surpassed for a detection of existing effects in the data—especially those that are small or moderate [Fie09]. Consequently, it may be that some of the tests that were conducted were not able to detect statistically significant impacts of gamification despite their actual existence. Second, the dataset that was obtained from the laboratory experiment contained nine responses (16%) with missing data. To address this problem multiple imputation was used to derive complete versions of the dataset. Although the results of all imputations coincided with those of the original, this is a limitation nevertheless as any restorative procedures introduce a certain degree of uncertainty. Third, whenever data is collected by means of a survey, it should be expected that some individuals respond in a careless manner or try to manipulate the outcome due to unknown reasons. To address this, the datasets in both experiments were scanned for outliers, which resulted in the deletion of a small number of responses in both cases. Notwithstanding these measures, it is still possible that some outliers were not detected and thus remain in the data. Last, despite the iterative nature of both design science and gamification (cf. Section 1.3), the core activities of the overall research process were only executed once due to the constraints imposed by the project conditions. Therefore, any insights gained from this evaluation have yet to be transformed into improvements to the gamification design concept and its implementation, and re-evaluated in an improved setting with enhanced measurement instruments.

8.4.3 Threats to Validity

Internal validity is concerned with whether the relationships observed in a study are truly caused by an experimental condition, or if they may be explained by alternative influence factors [SBG04]. If it is violated, then any reported findings are cast into doubt due to the uncertainty of their underlying reasons. For the studies at hand, the following potential threats discussed by Creswell are deemed to be relevant [Cre09]:

- *History*: "Because time passes during an experiment, events can occur that unduly influence the outcome beyond the experimental treatment"

[Cre09, p. 163]. This is only relevant for the field experiment, which yielded datasets for four semesters, and thus four different sets of students. To compensate for this, the analysis has focused on the comparison of subsets of data from the same semester. However, it can still not be perfectly ensured that all subjects were equally affected by events such as occasional server crashes or other technical problems, which may have affected their results. For the comparison of the two datasets of the experimental groups from 2015 and 2016, Section 8.2.1 has provided an extensive discussion narrowing down the main difference between these two semesters to the introduction of "soft goals". However, the existence of another, unobserved influencing factor cannot be rejected completely for this particular test.

- *Maturation*: "Participants in an experiment may mature or change during the experiment, thus influencing the results" [Cre09, p. 163]. This was not an issue for the laboratory experiment, which lasted less than an hour and thus presented little chance for maturation. In case of the field experiment, it can reasonably be assumed that most participants, being first-semester students, had similar, low levels of process modeling skills in the beginning. However, any divergence that may have occurred over the course of a semester was not observed. Nevertheless, it is reasonable to assume that most students developed their skills at a similar rate, and that even if this is not the case, any individual differences are randomly distributed among the experimental and control groups. Lastly, the existence of subjects with varying skills is one of the fundamental assumptions of the context for which the gamification module was designed, namely Social BPM. Thus, positive impacts should persist even when subjects with varying degrees of maturation exist.

- *Selection*: "Participants can be selected who have certain characteristics that predispose them to have certain outcomes" [Cre09, p. 163]. Clearly, the set of participants in both experiments may include students with superior, average, and also inferior capabilities and intelligence. Furthermore, as students who participated in the laboratory experiment were recruited from the field experiment, their skill sets can vary depending on the success of their participation in the lecture. To compensate for this, the assignment of students to the experimental or the control group was either conducted unknowingly by the students themselves, or in a randomized fashion, thereby increasing the probability that students with different characteristics are equally distributed among both groups.

- *Diffusion of treatment*: "Participants in the control and experimental groups communicate with each other[, which] can influence how both groups score on the outcomes" [Cre09, p. 163]. Through the design of the laboratory experiment, students were not aware of the existence of an experimental condition and their subdivision into two groups. However, some visual indications for this could be found on the questionnaire and in the user interface of the HBM, which only displayed gamification features for some. To compensate for this, the task was designed in a way so that it allowed for little slack, thereby decreasing the chance that students had enough time to consult the monitors of their peers. Furthermore, exam conditions were upheld during the execution of the experiment as far as possible. For the field experiment, no attempt was made to prevent the diffusion of treatment.

External validity is concerned with the extent to which the findings gained from a study can be generalized to other groups of individuals, settings, and timeframes [Cre09, Rec13]. If it is violated, then any results are only valid for the particular conditions under which an experiment was carried out (assuming internal validity), and may not be transferred to other contexts. For the studies at hand, the following potential threats discussed by Creswell are deemed to be relevant [Cre09]:

- *Interaction of selection and treatment*: "Because of the narrow characteristics of participants in the experiment, the researcher cannot generalize to individuals [with other characteristics]" [Cre09, p. 165]. The use of student subjects in IS research is a controversial issue with regard to the generalization of any findings to other groups of individuals [CMKH12]. As in many other studies, one of the main reasons for the collection of data from students is their ready availability in the academic environment. However, it is also believed that they can serve as an appropriate replacement for novice modelers as they may be found in manifestations of Social BPM. Specifically, both groups are expected to possess negligible process modeling skills, and may thus experience the support through gamification in a similar fashion. Nevertheless, student subjects are not an adequate substitute for modeling experts, although they may one day belong to this group themselves as working professionals. Therefore, under the assumption that expert users may be impacted by the features the gamification module provides in a different fashion, the results gained from the experiments cannot be generalized to this particular target group. Consequently, conducting additional experiments in a "real-world" setting involving working professional is a promising avenue for future research.

- *Interaction of setting and treatment*: "Because of the characteristics of the setting of participants in an experiment, a researcher cannot generalize to individuals in other settings" [Cre09, p. 165]. In Chapter 4, Social BPM was introduced as the setting in which the application of gamification to process modeling should be studied. Clearly, neither of the experiments that were conducted embodies the principles of social software, i. e., self-organization, egalitarianism, collective intelligence, and social production (cf. Section 4.1). However, many of the challenges that occur in Social BPM are not unique to that setting, and their relevance outside the former is evidenced by relevant literature. For instance, none of the empirical studies on model quality listed in Table 4.2 makes any assumptions about the nature of process management. Similar observations can be made for publications in the area of BPM discussing issues regarding motivation, education, and training. Thus, even though Social BPM unites all of these challenges under a common term, this does not mean that they are unique to the former. Consequently, it is argued that the results of this evaluation are valid for all types of BPM settings in which quality, motivation, and education pose a problem—including Social BPM. However, it may still hold that the impacts of gamification manifest themselves differently in the presence of BPM practices following the principles of social software, and thus the execution of additional studies in more "faithful" settings is considered future work.

Part III

Closing Remarks

9 Conclusions

To conclude this thesis, Section 9.1 briefly recapitulates its underlying motivation and provides a concise summary of its most significant outcomes. Then, Section 9.2 proceeds to discuss the implications of its results for research and practice. Finally, the most promising opportunities to continue the work presented in this thesis are laid out in Section 9.3.

9.1 Summary

For a long time, play has been considered a purely voluntary activity without lasting consequences in the real world. Hence, games were seen as entirely unproductive and unable to serve any "serious" purpose beyond mere entertainment. Today, this view has become obsolete, and researchers and practitioners alike have begun to harness the powers of games to produce tangible results outside the "magic circle" of play. Simultaneously, our society is continuously becoming more playful, meaning that more and more individuals can be reached through approaches that are grounded in digital games. As a result, many different research directions have emerged, such as serious games, games with a purpose, game-based learning, pervasive games, and gamification. The latter is concerned with the use of game design elements in non-game contexts such as education, health, or work, and is one of the most recent manifestations of this type of research. Today, gamification has surpassed the status of being the newest "buzzword", and research in this area is continuing to mature and increase in sophistication despite facing some resistance and criticism.

In this thesis, the application of gamification to business process modeling was examined. While the impacts of gamification have already been extensively studied in a large variety of different non-game contexts, this is not the case for the selected domain of interest, and thus little related work to build on could be identified. This resulted in the need to first identify the potential contributions of gamification to business process modeling. Therefore, the following high-level, exploratory research question was defined in the introduction: *how can gamification be used to support business process modeling*? To examine this question, design science research was conducted to identify relevant open problems of business process modeling that can be addressed through the creation of an appropriate IT artifact, i. e., a gamified process modeling tool. In accordance with this aim, the contents of this thesis were subdivided into the following two parts.

Part I presented the foundations on which the remainder of the thesis was built, namely business process management and gamification. For the former, Chapter 2 first defined and characterized some of the most important key terms in a bottom-up fashion. Specifically, the following concepts were addressed: business processes, business process models, business process modeling, and business process management. However, the most important outcome of this chapter is an extensive list of over 30 quality metrics which quantify particular aspects of the quality of a process model. These metrics address numerous quality concerns, such as the visual representation of a process model, its inherent complexity, the format of its textual contents, and its semantics. Afterwards, the discussion of gamification in Chapter 3 starts with a synthesis of various definitions of the term, thereby extracting the essential characteristics of this novel approach. Furthermore, important gamification examples from practice and academia are presented to enhance the tangibility of its underlying principles. Additionally, related concepts are outlined to delineate gamification from other approaches grounded in games. This is followed by the key outcomes of this chapter, which are twofold. Firstly, an extensive overview of the broad array of game design elements that gamification designers can operationalize is provided. This overview includes game interface design patterns, game design patterns and mechanics, game design principles and heuristics, game models, and game design methods. Secondly, six important theories that are commonly used to explain the underlying mechanisms of gamification are outlined. The majority of these theories are concerned with motivation and address different factors that influence human behavior. The contents of these chapters were crucial ingredients for the IT artifact that was created and presented in detail in Part II, but also provide comprehensive information on the two domains they address that goes beyond the immediate requirements of this thesis.

Part II followed the key phases of the design science process underlying this thesis to document how an IT artifact for gamified business process modeling was designed, implemented, and evaluated. The resulting artifact manifested itself as a gamification module that was integrated into the Horus Business Modeler (HBM), a commercially available software program for business process modeling. To arrive at a set of concrete goals for the artifact, Chapter 4 first shifted the focus from "traditional" business process management to Social BPM, which is concerned with establishing an "architecture of participation" for BPM by adopting the underlying principles of social software. This is intended to enable and encourage relevant process stakeholders to contribute their domain knowledge and method expertise to all activities of the BPM life cycle. By means of a literature review, the following three challenges of Social BPM were identified and the potential contributions of gamification outlined: ensuring high levels of model quality while allowing the active involvement of modeling novices, ensuring the participation of all relevant process stakeholders, and providing appropriate tools for training and education. After this, Chapter 5 outlined the technical and organizational context in which the developed artifact is embedded. Specifically, this research is the result of an ongoing collaboration between the DBIS Group and the Horus software GmbH situated in Karlsruhe. Therefore, it had to be ensured that its outcomes were compatible with the Horus method and the HBM developed, marketed, and utilized by the Horus software GmbH.

In Chapter 6, the design concept for the gamification module of the HBM was showcased. To that extent, a variety of game design elements and other features relating to tasks, rewards, players, competition, interaction, and resources were described and illustrated through examples from games or gamified applications. Furthermore, design lenses were introduced as a tool that allows looking at a gamification design from different perspectives to emphasize certain components that are expected to effectuate particular results. Specifically, five lenses corresponding to the three project goals presented in Chapter 4 were examined. Of these lenses, one was focused on model quality, three on motivation, and the last one on training and education. Using this tool, hypotheses about the gamification concept were implicitly formulated in preparation for the evaluation phase. Next, Chapter 7 demonstrated the conversion of the design concept into the implementation of a gamification component for the HBM. To that extent, the chapter first shed light on supporting features that are not directly related to gamification, but are essential for enabling the latter. These features are concerned with tracking user behavior, deriving quantitative indicators for user performance, and quantifying the quality of business process models based on the considerations recorded in Part I. Next, the chapter illustrated the technical realization of a subset of the features presented in the design concept. In particular, the following game design elements and features to support the former were realized: a status panel and notification area, a user profile, a points system, a badge system, a leaderboard, and a dashboard with an introduction component. These features cover roughly one third of all elements in the concept and are most strongly associated with the project goal of maintaining high model quality.

Chapter 8 was concerned with assessing whether the expected positive impacts actually manifested through an empirical evaluation. To that extent, two experiments were carried out with the support of Bachelor-level IS students at the University of Münster: a field experiment integrated into the lecture *Introduction to IS*, and a laboratory experiment based on a mock exam whose participants were recruited from the former. From these experiments, multiple datasets with quantitative information about the quality of the created models, actual user behavior, user experience, flow, intrinsic motivation, extrinsic motivation, technology acceptance, and self-efficacy were obtained. Through the application of appropriate statistical instruments, several findings could be derived from the collected data. Most notably, it was shown that gamification can have a positive impact on model quality if utilized appropriately, and that the integration of game design elements into a process modeling tool leads to a better assessment of some aspects related to user experience and technology acceptance. Most notably, students who worked with the gamified version of the HBM expressed higher intentions to continue using the tool in the future despite the fact that its use was mandated rather than voluntary. Surprisingly, the same cannot be said for the examined motivational concepts, for which the use of gamification did not yield any statistically significant outcomes. However, prior research in other contexts has shown that gamification can indeed have a positive impact on motivation, which indicates that future work must look into the reasons for its ineffectiveness in the setting presented here in more detail.

In conclusion, the goal of this thesis was the identification of a relevant organizational problem, and contributing towards its resolution through the creation of an appropriate IT artifact. To that extent, a set of three challenges that are of particular significance for business process modeling in the context of Social BPM—a democratized, bottom-up approach to process management—was discussed. Furthermore, a design concept for a gamified business process modeling tool was proposed, and its instantiation in terms of a concrete implementation presented. The results of the empirical evaluation show that gamification can indeed yield measurable, statistically significant benefits for process modeling, even though it was not possible to confirm all hypothesized relationships. Nonetheless, the research question introduced at the beginning of this thesis was answered, and thus the work presented here can be considered a success. Even so, promising opportunities for future research remain and are discussed in the following two sections.

9.2 Implications

The findings presented in this thesis have implications for BPM research, research on gamification, BPM practitioners, and vendors of business process modeling tools. The most significant consequences for these parties are outlined in the following paragraphs.

Implications for BPM research. The research presented in this thesis illustrates how business process modeling can be designed as a gameful activity whose ultimate goal it is for players to create high-quality process models. Building on the findings presented in Chapter 8, researchers in this field may expand upon this work in the following ways. First, the independent variable may be altered while leaving the dependent variables unchanged. This would mean selecting (and possibly first developing) different approaches for the goals defined in Chapter 4, and studying what kind of artifact provides the best results under which conditions. Focusing on model quality, alternatives could for instance be merely providing modelers with verbal explanations of quality goals, or supplying them with delayed rather than real-time quality feedback. Second, researchers may leave both the independent and dependent variables unchanged, but develop different gamification designs and implementations for process modeling. As mentioned in Chapter 1, creating a gamified application is a highly creative process with an unbounded number of possible solutions for any given problem, and thus results can be expected to change for different artifacts. By analyzing the impacts that varying designs have on the dependent variables, it may be possible to determine which game design elements yield the best outcomes in which situations. Third, the independent variable may be left unchanged while choosing a different set of dependent variables to measure. Here, this would mean using the gamification module implemented for the HBM as described in Chapter 7, and designing experiments for assessing its impacts on different constructs than those employed in the evaluation, or measuring the same constructs using different instruments (e. g., assessing flow using the DFS-2 rather than the FKS). Fourth, the underlying principles of this research may also be applied to other types of conceptual models, such as data models or enterprise architecture models. Here, it may be necessary to vary certain aspects of the design concept, such as the set of quality metrics, to meet the requirements of the particular domain. Finally, while the research presented here is concerned with business process modeling, the BPM life-cycle is also comprised of many additional activities that may be gamified as well. Consequently, this work could motivate other researchers to study novel topics at the intersection between games and BPM, such as gamified process mining, gamified process monitoring and evaluation, or model-driven, gamified business process execution. Ultimately, this may result in the notion of a gamified BPM life-cycle that unites the various contributions that gamification can make in this domain.

Implications for gamification research. The outcomes of this thesis can inform gamification research in multiple ways. First, the findings presented in Chapter 8 extend the body of empirical research in this field, and thus provide academics with additional insights into when, how, and why gamification does and does not work. For instance, the results of the field experiment and the laboratory experiment both suggest that the impacts of gamification do not only depend on its implementation, but also contextual factors such as task difficulty and the manner of goal-setting. Eventually, this may contribute to the creation of gamification-specific theories, and to the advancement of the research discipline as a whole. Second, the laboratory experiment that was presented in Chapter 8 differs from the majority of previous empirical gamification research in that it incorporates validated, psychometric measurement instruments (most notably the intrinsic motivation inventory), and in that it presents outcomes that are mostly negative. The former is of interest since researchers commonly cite motivational theories such as those presented in Section 3.5 as possible explanations for the impacts of gamification, but do not utilize measurement instruments that are appropriate for investigating the hypothesized relationships. The latter is of special note, as research in which the proposed connections cannot be confirmed is often perceived as a failure, and thus not reported. However, it is reasonable to assume that researchers may learn as much from negative results as from positive findings. Lastly, due to space constraints, many publications describing applied gamification research provide very few details about their design and implementation procedures, and instead limit their focus to the evaluation of the created artifacts. While this is undeniably important, gamified applications are often complex combinations of pre-existing functionality and numerous elements with different characteristics. Therefore, the detailed documentation of design and implementation in this thesis might enable other researchers to gain a deeper understanding of the causal chain that links goals, design, implementation, and results in gamification research.

Implications for practice. For managers involved in decision-making about the BPM practices of their organization, the research presented in this thesis offers various potentials. First, it contributes towards the feasibility of Social BPM by introducing gamification as an approach through which some challenges of democratized, bottom-up process management can be addressed. Most significantly, the design concept demonstrates how game design elements may motivate process stakeholders to participate in BPM activities, facilitate the maintenance of model quality despite the active participation of novice users in process modeling, and provide measures for training and education. Furthermore, the evaluation results presented in Chapter 8 offer tangible proof for a subset of these claims. On this basis, managers can adjust their decision-making about Social BPM and make a well-educated ruling about the acquisition of a gamified business process modeling tool. Second, Chapter 2 includes an extensive (albeit not exhaustive) overview of many topics related to model quality, thereby educating practitioners about different characteristics that a “good” process model should have and how they can be quantified. Finally, the results of the evaluation highlight which of the associated quality metrics novice modelers find particularly difficult to operationalize (i. e., which metrics exhibit the lowest average values), which allows focusing the search for quality defects in larger process model repositories on issues that can be expected to be present with a higher likelihood. For vendors of business process modeling software, the design concept presented in Chapter 6 offers an extensive list of different features through which gamification can be integrated into an already-existing modeling tool independently of the underlying modeling language. By using the five presented design lenses, vendors may further choose a subset of game design elements to selectively address either model quality, motivation, or training and education rather than having to implement the entire concept. Additionally, the description of the implementation in Chapter 7 provides insights into the technical challenges of implementing such features and outlines architectural choices that allow overcoming the former. For instance, the chapter illustrates how a sophisticated reward system for points and badges can be built bottom-up from basic event tracking capabilities.

9.3 Outlook

While a considerable amount of time and effort was invested in the creation of the gamification module of the HBM, many possibilities to continue the research presented in this thesis exist. The most significant opportunities for future work are presented in the following paragraphs.

Implementation of further quality metrics. In Section 2.7 of this thesis, over 30 different types of quality metrics addressing issues related to planar variables, retinal variables, model complexity, textual model contents, and model semantics were introduced and described in detail. To examine the impacts of gamified real-time model quality feedback, it was not necessary to implement all of these metrics in the first version of the HBM gamification module, and thus the implementation presented in Chapter 7 only realizes a subset of 18 metrics[1] satisfying certain selection criteria. Thus, one promising avenue for future research lies in the implementation of additional metrics to provide more comprehensive feedback for various aspects of model quality. In

[1] The total number of provided metrics is 32; however, all completeness metrics as well as both metrics related to consistent flow are realizations of the same concept, and thus only counted once.

this context, it would be of special interest to not only consider metrics that are concerned with structural aspects of process models, but to also examine possibilities to quantify the quality of the subject matter that is depicted. Issues of this nature are related to, e. g., the semantic, social, and deontic quality levels of the SEQUAL framework, which the current implementation either addressed in a rudimentary fashion, or not at all.

Extension of the implementation. Due to the extent of the gamification design concept presented in Chapter 6 and time and resource constraints, only a subset of the concept was implemented and evaluated. From a purely enumerative point of view, 61% of the proposed game design elements have yet to be realized, and while the implementation provides adequate support for some of the design lenses outlined in Section 6.2, this is not the case for others. Some of the most notable elements that the gamification module is currently missing are related to team tasks and the research goal of providing gamified measures for training and educating the participants of Social BPM. Consequently, implementing the remaining parts of the gamification concept and examining their impacts on quality, motivation, and training is also considered future work. Some initial work to that end was carried out in the context of a student project supervised by the author of this thesis. In this project, a prototype of the gamification module extended with tutorials, quizzes, a skill tree, hidden badges, an experience points and level system, and user privileges as an additional type of rewards was created. This prototype clearly illustrated the technical feasibility of the proposed features, and it is planned to continue the collaboration between the DBIS Group and the Horus software GmbH in this particular direction with the support of a new doctoral student.

Iteration of the research process. An inherent characteristic of design science research as well as gamification design lies in the iterative nature of its associated processes. Consequently, it is not required to create the best-possible solution to a given problem already in the first attempt. Instead, it is of particular importance to continuously assess and evaluate the performance of the created artifacts to determine potentials for their improvement. For instance, the results presented in Chapter 8 illustrate the importance of goal-setting for quality improvement, and thus the design concept may be adapted to provide more sophisticated facilities for this purpose. Additional findings demonstrate that a proper balance between the skill sets of modelers and the difficulty of the tasks they are trying to accomplish might be a precondition for intrinsic motivation and the occurrence of flow experiences. Therefore, it might be necessary to provide modelers with support in choosing tasks that match their current level of expertise. Overall, these (and further) insights should be fed back into a second iteration of the main research phase to improve the gamification design concept and its corresponding implementation.

Improvement of the evaluation. While the evaluation of the gamification module presented in Chapter 8 allowed detecting statistically significant effects of the gamification module, some potentials for improving the evaluation in future iterations of the research process remain. First, the sample sizes of both experiments did not meet the recommendations commonly found in the literature, which increases the difficulty of assessing validity and detecting significant effects of small or medium size. Therefore, it should be attempted to increase the size of the sample for future evaluations. Second, due to some shortcomings in data collection, some data could not be considered in the evaluation despite its availability. This most notably affects data for the field experiment about system use in 2016/17, as there was no possibility to assign individual records to either the experimental group or the control group. Consequently, it must be ensured that this distinction can always be made in future experiments. Third, the utilized measurement instruments exhibited some weaknesses, and thus should be adapted accordingly. For instance, it may be advisable to evaluate flow with the DFS-2 rather than the FKS, to reduce the overall number of questions, and to abandon the CUE model in favor of the TAM, which is more widely-used in IS research. Fourth, the results of the laboratory experiment clearly demonstrate that students perceived the modeling task as too difficult, which may have had a negative impact on certain motivational constructs. Therefore, it should be attempted to reduce the difficulty level of the task so that it is more closely aligned with the capabilities of the participants. Finally, it may be advisable to conduct future experiments in real-world settings involving working professionals and actual Social BPM to improve the external validity of results. Nevertheless, it is maintained that the findings presented in Chapter 8 are valid despite the use of student subjects for data collection.

Part IV

Appendix

A Concept Details

This chapter of the appendix provides detailed information about certain aspects of the gamification design concept presented in Chapter 6. Specifically, Section A.1 outlines the completion logic of the set of badges following the badge description framework proposed in [HE11], whereas Section A.2 specifies the intended primary purposes of these badges according to the classification defined in [AC11]. Furthermore, Section A.3 presents the lower and upper bounds that were used for quality metrics.

A.1 Badge Completion Logic

Based on the badge design framework presented in [HE11], Section 6.1.2 has described the *signifiers* of the 31 badges included in the Horus gamification concept. For each badge, Table 6.2 listed a numeric identifier, the badge level, its name, description, and a visual icon. While the descriptions of all badges already indicate which actions users must perform to unlock the respective achievement, Hamari and Eranti suggest defining the *completion logic* of all badges in an unambiguous manner [HE11]. To that extent, Table A.1 lists the following information: the *trigger* that must occur for an examination of the badge reward condition, the *condition(s)* that must be met for the badge to be unlocked, and the *multiplier* indicating how often the trigger has to occur with satisfied condition(s). From the data contained in Table A.1, reward conditions such as those presented as part of the implementation in Section 7.4.4 can directly be derived.

Table A.1: List of badges in the Horus gamification concept: Completion logic.

ID	Level	Trigger	Condition	Multiplier
Completion of user profile				
1	①	Profile update	Must contain at least first name, last name	1
	②	Profile update	Must contain first name, last name, email, description	1
2	①	Profile update	Must contain profile image	1
3	①	Skill addition	Skill must not be removed	5
4	①	Contact method addition	Contact method must not be removed	5
5	①	Skill endorsement	Endorsement must not be removed	10
General user activity				
6	①	Login by user	No further condition	1
	②	Login by user	No further condition	100
7	①	Login by user	Logins must have occurred on subsequent days	3
	②	Login by user	Logins must have occurred on subsequent days	5
	③	Login by user	Logins must have occurred on subsequent days	7
8	①	Login by user	Account age in years ≥ 1	1

Continued on the next page ▷

Table A.1 (contd.)

ID	Level	Trigger	Condition	Multiplier
	②	Login by user	Account age in years ≥ 2	1
	③	Login by user	Account age in years ≥ 3	1
9	①	Login by user	Login date must be 2015/12/24	1
10	①	Login by user	Login date must be 2016/01/01	1
Activities in the workspace explorer				
11	①	Creation of process model	No further condition	10
	②	Creation of process model	No further condition	100
12	①	Creation of (any) model	No further condition	10
		Deletion of (any) model	No further condition	5
13	①	Creation of (any) model	No further condition	5
		Deletion of (any) model	No further condition	10
14	①	Creation of model	At least process, object, organization model and role	1
	②	Creation of model	At least one of each type of model	1
Process modeling				
15	①	Removal of edge crossing	Model changes must be saved	10

Continued on the next page ▷

Table A.1 (contd.)

ID	Level	Trigger	Condition	Multiplier
	②	Removal of edge crossing	Model changes must be saved	100
16	①	Addition of object type to object store	Model changes must be saved	30
17	①	Addition of role to activity	Model changes must be saved	30
18	①	Addition of character of textual description	Model changes must be saved	10,000
19	①	Change of model diameter	Model changes must be saved; diameter≥30	1
20	①	Change of model size	Model changes must be saved; size≥50	1
21	①	Displacement of model element by a pixel	Model changes must be saved	1,000,000
22	①	Receipt of a gamification point	No further condition	10,000

A.2 Badge Purposes

In gamification literature, a "deterministic one-to-one relationship" [Det14, p. 317] between achievements and extrinsic motivation is often assumed. However, as Deterding points out in an argument based on the work by Antin and Churchill, even a simple game design element such as a single badge can impact different players in many different ways [Det14]. Specifically, the positive effects of badges can manifest themselves in at least five different ways (repeated from Section 3.4.1) [AC11]:

- *Goal setting*: In this function, badges represent a challenge for users to overcome. Such goals can be a strong motivator and play an important role in several theories referred to by gamification publications, such as Flow theory and Goal-setting theory (see Section 3.5). Research indicates that badges are most effective when the goals they represent are achievable and users receive feedback on their progress. Consequently, it has been observed that users exert more effort when they are close to unlocking a badge [MK14].

- *Instruction*: Through their reward conditions, some badges provide users with information about which actions are possible in the gamified system and, more importantly, which actions represent highly-valued behavior. Thus, badges can help with shaping user activity into the desired direction. Furthermore, the ability to view a full list of available badges enables users to gain a holistic understanding of the gamified domain.
- *Reputation*: Badges can provide information about the interests, skills, expertise, and behavior of the users who have obtained them. Thereby, they allow assessing the reputation of a user, help with assessing the trustworthiness and reliability of any content produced by the user of a gamified system, and can serve as a substitute for direct interaction.
- *Status and affirmation*: Badges can serve as a status symbol that allows users to communicate their accomplishments to others. In this context, the expectation of how a badge is perceived by others is more important than its actual impact. Besides status, badges may also reaffirm users by reminding them of past milestones they have achieved.
- *Group identification*: By means of their reward conditions, badges define subsets of users who have undergone the same trial and thus share certain experiences. Through this, badges may cause a sense of positive identification and solidarity within the respective group of users.

Based on this classification, Table A.2 provides a list of all badges in the Horus gamification design concept (see Section 6.1.2) and their intended primary purposes. While many badges may serve more than one purpose at the same time (e. g., the badge “Object Type Completionist” provides a concrete goal for modelers, but also instructs them about desired behavior that should persist beyond the goal), only one purpose is addressed for reasons of simplicity. Furthermore, a detailed argumentation for each badge is beyond the scope of this thesis, and thus no details for the asserted relationships are given.

Table A.2: List of badges in the Horus gamification concept with intended primary purpose. GoSe: Goal setting, Inst: Instruction, Repu: Reputation, StAf: Status and Affirmation, GrId: Group identification.

ID	Level	Name	GoSe	Inst	Repu	StAf	GrId
Completion of user profile							
1	①	I know your name	✔	✖	✖	✖	✖
	②	I know where you live!	✔	✖	✖	✖	✖
2	①	Look at me!	✔	✖	✖	✖	✖
3	①	Reachable	✖	✔	✖	✖	✖
4	①	Capable	✖	✔	✖	✖	✖
5	①	Endorser	✖	✔	✖	✖	✖
General user activity							
6	①	New User	✖	✔	✖	✖	✖
	②	Power User	✖	✖	✔	✖	✖
7	①	Returner	✖	✖	✔	✖	✖
	②	Metronome	✖	✖	✔	✖	✖
	③	Junkie	✖	✖	✔	✖	✖
8	①	It's your Birthday	✖	✖	✖	✔	✖

Continued on the next page ▷

Table A.2 (contd.)

ID	Level	Name	GoSe	Inst	Repu	StAf	GrId
	②	It's your Birthday... again	✖	✖	✖	✔	✖
	③	It's your Birthday... yet again	✖	✖	✖	✔	✖
9	①	Christmas Modeler 2015	✖	✖	✖	✖	✔
10	①	New Year's Modeler 2016	✖	✖	✖	✖	✔
Activities in the workspace explorer							
11	①	Process Model Creator	✔	✖	✖	✖	✖
	②	Process Model Creator	✔	✖	✖	✖	✖
12	①	Amateur Constructivist	✖	✖	✔	✖	✖
13	①	Amateur Destructivist	✖	✖	✔	✖	✖
14	①	Zen Novice	✖	✖	✔	✖	✖
	②	Zen Master	✖	✖	✔	✖	✖
Process modeling							
15	①	You Shall Not Cross	✖	✔	✖	✖	✖
	②	You Shall Not Cross	✖	✔	✖	✖	✖
16	①	Object Type Completionist	✖	✔	✖	✖	✖

Continued on the next page ▷

Table A.2 (contd.)

ID	Level	Name	GoSe	Inst	Repu	StAf	GrId
17	①	Role Completionist	✖	✔	✖	✖	✖
18	①	Babbler	✖	✔	✖	✖	✖
19	①	Snake Man	✖	✔	✖	✖	✖
20	①	Megalomaniac	✖	✔	✖	✖	✖
21	①	Pixel Pusher	✖	✖	✖	✖	✔
22	①	Points Collector	✖	✖	✖	✔	✖

A.3 Quality Metric Bounds

Table A.3 depicts the lower and upper bounds that were used for quality metrics in the design concept and realized in the implementation. Some of these bounds deviate from the recommendations outlined in Section 2.7, as they were chosen together with the project partner, occasionally set before the examination of suggested bounds in the literature, or found to be more appropriate for gamified process modeling through experimentation.

Table A.3: Bounds of the quality metrics in the Horus gamification concept.

Metric	Lower Bound	Upper Bound
Readability		
Edge Crossings	0	Maximum number of crossings
Edge Bends	1 per arc	1.5 per arc
Node Occlusion	0	Number of nodes
Angular Resolution	0	Number of (promoted) arcs
Consistent Flow 1	0	Number of arcs
Consistent Flow 2	0	Number of arcs
Orthogonality	0	Number of (promoted) arcs
Complexity		
Size	15	25
Diameter	10	20
Density	0	1
Connectivity Coefficient	1	3
Avg. Connector Degree	2	6
Max. Connector Degree	2	6
Connector Mismatch	0	Number of possible mismatches
Control Flow Complexity	3	10
Cyclicity	0	Number of places and transitions
Token Split	0	5
Sources and Sinks	2	4
Completeness		
Names	0	Number of nodes
Short Names	0	Number of nodes
Descriptions	0	Number of nodes
Notes	0	Number of nodes
Business Rules	0	Number of transitions
Documents	0	Number of nodes
KPIs	0	Number of transitions
Object Types	0	Number of places
Refinements	0	Number of transitions
Resources	0	Number of transitions
Risks	0	Number of transitions
Roles	0	Number of transitions
Services	0	Number of transitions
System Components	0	Number of transitions

B Implementation Details

This chapter of the appendix provides detailed information about certain aspects of the implementation of the HBM gamification module presented in Chapter 7. Specifically, Section B.1 lists all types of events that are currently tracked by the implementation. Furthermore, a full list of all performance metrics that are currently computed is given in Section B.2. The chapter ends with Java source code extracts demonstrating the implementation of a performance measure and a quality metric in Section B.3 and Section B.4, respectively

B.1 Tracked Events

This section provides a full list of the events that are currently tracked by the Horus Business Modeler. For each type of event, its unique textual identifier and a short description are given. Further information about the data that is stored for each event type (cf. Table 7.1) is not indicated, as such implementation details are not required for the scope of the thesis. To structure this section, events that are similar with regard to their origin and meaning are grouped together.

Workspace events This type of event is related to changes in the set and structure of workspaces that exist in a particular repository. All events of this type have the common prefix events.projectManager., which is omitted in the following list.

Table B.1: Tracked events: Workspace events.

Event Type	Description
workspaceCreated	*User created a new workspace.*
workspaceDeleted	*User deleted a workspace.*
workspaceRenamed	*User renamed a workspace.*
workspaceMoved	*User moved a workspace.*
workspaceCreatedAsCopy	*User copied a workspace.*

Model events This type of event is related to changes in the set and structure of models that exist in a particular repository. All events of this type have the common prefix events.projectManager., which is omitted in the following list.

Table B.2: Tracked events: Model events.

Event Type	Description
modelCreated	*User created a new model.*
modelDeleted	*User deleted a model.*
modelRenamed	*User renamed a model.*
modelMoved	*User moved a model.*
modelCreatedAsCopy	*User copied a model.*

Editor events This type of event is related to the user interacting with models and their respective model editors. All events of this type have the common prefix events.editor., which is omitted in the following list.

Table B.3: Tracked events: Editor events.

Event Type	Description
modelOpened	*User opened a model in its editor.*
modelClosed	*User closed the editor of a model.*
modelSaved	*User saved changes made to a model.*

Quality events This type of event is related to the quality of models as measured by quality metrics. All events of this type have the common prefix events.quality., which is omitted in the following list.

Table B.4: Tracked events: Quality events.

Event Type	Description
changed	*Quality of a model has changed*

User events This type of event is related to changes in the status of the user, changes the user is making to his own user profile, or interactions of the user with the profiles of others. All events of this type have the common prefix events.user., which is omitted in the following list.

Table B.5: Tracked events: User events.

Event Type	Description
contact.added	*User added contact information (e. g., e-mail, phone).*
contact.removed	*User deleted contact information.*
skill.added	*User added a skill he possesses.*
skill.removed	*User deleted a skill he possessed.*
skill.endorsed	*User endorsed another user's skill.*
skill.unendorsed	*User stopped endorsing another user's skill.*
login	*User logged into the repository.*
logout	*User logged out of the repository.*
pic.uploaded	*User uploaded a new profile picture.*
profile.changed	*User changed his profile information (e. g., name).*

Help events This type of event is related to interactions of the user with the gamification help pages of the HBM. These events are used to present the user with a checklist of initial activities to perform when starting to use the software. All events of this type have the common prefix events.help.read., which is omitted in the following list.

Table B.6: Tracked events: Help events.

Event Type	Description
businessProcesses	*User read help page about business processes.*
gamiUseOrga	*User read help page about orga. gamification usefulness.*
gamiUse	*User read help page about individual gamification usefulness.*
gamification	*User read help page about gamification.*
petriNets	*User read help page about Petri nets.*
rewards	*User read help page about rewards that can be earned.*
learning	*User read help page about what can be learned.*

Gamification events This type of event is related to outcomes of using gamification functionality, i. e., earning rewards. All events of this type have the common prefix events.reward., which is omitted in the following list.

Table B.7: Tracked events: Gamification events.

Event Type	Description
points.received	*User has received a specific amount of points.*
badge.unlocked	*User has unlocked a particular badge.*

Petri net element creation events This type of event is related to the creation of elements in the Petri net editor. Elements are either model elements (i. e., they define the execution semantics of a process model) or graphical elements (i. e., they only serve visualization purposes and do not have an execution semantics). All events of this type have the common prefix events.editor.petriNet.created., which is omitted in the following list.

Table B.8: Tracked events: Petri net element creation events.

Event Type	Description
model.arc	*User created an arc specifying control flow.*
model.place	*User created an object store/place.*
model.transition	*User created an activity/transition.*
graphics.note	*User created a graphical note.*
graphics.image	*User created a graphical image.*

Continued on the next page ▷

Table B.8 (contd.)

Event Type	Description
graphics.freeText	*User created a graphical free text element.*
graphics.freeLine	*User created a graphical free line element.*
graphics.ellipsis	*User created a graphical ellipsis.*
graphics.rectangle	*User created a graphical rectangle.*
graphics.rectangle3d	*User created a graphical three-dimensional rectangle.*
graphics.rectangleRound	*User created a graphical rounded rectangle.*
graphics.cylinder	*User created a graphical cylinder.*
graphics.triangle	*User created a graphical triangle.*
graphics.diamond	*User created a graphical diamond.*
graphics.pentagon	*User created a graphical pentagon.*
graphics.hexagon	*User created a graphical hexagon.*
graphics.octagon	*User created a graphical octagon.*
graphics.freeShape	*User created a graphical free shape.*

Petri net element deletion events This type of event is related to the deletion of elements in the Petri net editor. All events of this type have the common prefix events.editor.petriNet.deleted., which is omitted in the following list.

Table B.9: Tracked events: Petri net element deletion events.

Event Type	Description
model.arc	*User deleted an arc specifying control flow.*
model.place	*User deleted an object store/place.*
model.transition	*User deleted an activity/transition.*
graphics.note	*User deleted a graphical note.*
graphics.image	*User deleted a graphical image.*
graphics.freeText	*User deleted a graphical free text element.*
graphics.freeLine	*User deleted a graphical free line element.*
graphics.ellipsis	*User deleted a graphical ellipsis.*
graphics.rectangle	*User deleted a graphical rectangle.*
graphics.rectangle3d	*User deleted a graphical three-dimensional rectangle.*
graphics.rectangleRound	*User deleted a graphical rounded rectangle.*
graphics.cylinder	*User deleted a graphical cylinder.*
graphics.triangle	*User deleted a graphical triangle.*
graphics.diamond	*User deleted a graphical diamond.*
graphics.pentagon	*User deleted a graphical pentagon.*

Continued on the next page ▷

Table B.9 (contd.)

Event Type	Description
graphics.hexagon	*User deleted a graphical hexagon.*
graphics.octagon	*User deleted a graphical octagon.*
graphics.freeShape	*User deleted a graphical free shape.*

Petri net graphical property change events This type of event is related to changes of graphical properties of the elements in a Petri net diagram and its visualization in the editor window. All events of this type have the common prefix events.editor.petriNet.propertyChanged.graphics., which is omitted in the following list.

Table B.10: Tracked events: Petri net graphical property change events.

Event Type	Description
addVirtualTokens	*Tokens added to transition postset in token game.*
alphaWert	*Transparency/opacity of an element changed.*
arrowStyle	*Head style of an arrow/arc changed.*
arrangeNodes	*Automatic model layout command invoked.*
bendpoint	*Arc bendpoint created or removed.*
backgroundColor	*Background color of an element changed.*
bordering	*Place has become linked/unlinked to/from other model.*
capacity	*Capacity of place has changed*
changeReferencing	*(Never fired.)*
childAdded	*New element added to model.*
childRemoved	*Existing element removed from model.*
cornerRadiusHeight	*Corner radius height of rounded rectangle changed.*
cornerRadiusWidth	*Corner radius width of rounded rectangle changed.*
defaultImage	*Default image of a model element changed.*
diagramName	*Title and file name of model changed.*
document	*List of files or URLs associated with element changed.*
enabled	*Transition has become enabled or disabled.*
endArrowFill	*Fill option of end arrow of free line changed.*
endArrowStyle	*Style of end arrow of free line changed.*
endArrowX	*Size of end arrow of free line changed.*
endArrowY	*Size of end arrow of free line changed.*
expanded	*Transition assoc. with subdiagram was (un)expanded.*
foregroundColor	*Foreground color of an element changed.*

Continued on the next page ▷

Table B.10 (contd.)

Event Type	Description
fill	*Fill color of an element changed.*
finish	*Free line with multiple segments was finished.*
firing	*Transition has started or stopped firing.*
font	*Font of a label was changed.*
gradient	*Fill color type of an element changed to gradient.*
gradientColor	*Fill color of gradient changed.*
gradientDirection	*Fill direction of gradient changed.*
gridColor	*Color of grid in model editor changed.*
gridEnabled	*Snap-to-grid option for model elements changed.*
gridSize	*Size of grid in model editor changed.*
gridVisible	*Visibility of grid in model editor changed*
height	*Height of a model element changed.*
horizontalAlignment	*Horizontal alignment of a text label changed.*
image	*Image of an image element changed.*
imageBorderStyle	*Border style of an image element changed.*
imageLabels	*Label with image icon (e. g., for roles) added or removed.*
incomingArc	*Incoming arcs of an element changed.*
lastLineWidth	*Previous line width of an arc changed.*
lineStyle	*Line style of an arc changed.*
lineWidth	*Line width of an arc changed.*
location	*Location of an element changed.*
mouseMove	*Mouse cursor moved (not processed or stored.s)*
outgoingArc	*Outgoing arcs of an element changed.*
pointList	*List of points of a freehand shape changed.*
removeVirtualTokens	*Tokens removed from transition preset in token game.*
router	*Element router in editor view added or removed.*
rulerUnits	*Unit for element routers was changed.*
rulersVisible	*Visibility of element routers was changed.*
scale	*Scaling factor of freehand shape changed.*
schemaPath	*XML schema associated with place changed.*
showDescription	*Show description option of an element changed.*
showName	*Show full name option of an element changed.*
showNotes	*Show notes option of an element changed.*
showViewName	*Show short option name of an element changed.*
size	*Size of an element changed.*
sizeLock	*Aspect ratio of freehand shape locked or unlocked.*
snapToGeometryEnabled	*Snapping of elements to other elements en-/disabled.*
startArrowFill	*Fill option of start arrow of free line changed.*

Continued on the next page ▷

Table B.10 (contd.)

Event Type	Description
startArrowStyle	*Style of start arrow of free line changed.*
startArrowX	*Size of start arrow of free line changed.*
startArrowY	*Size of start arrow of free line changed.*
subdiagram	*(Never fired.)*
text	*Textual contents of a label changed.*
tokenCount	*Number of tokens in a place changed.*
viewName	*Display/short name of an element changed.*
virtualTokenCount	*Number of tokens in a place during token game changed.*
visible	*Visibility of an element changed.*
width	*Width of a model element changed.*
xorConnection	*Status of arc regarding XOR/exclusivity changed.*
xPos	*Position of an element on the X axis changed.*
yPos	*Position of an element on the Y axis changed.*
cust.arcType	*Type of arc (normal, read, update) changed.*
cust.fill	*Element color fill option changed.*
cust.imageAlignment	*Alignment of image changed.*
cust.inAndOutputOR	*Transition set as input XOR and output XOR.*
cust.inputOR	*Transition set as input XOR.*
cust.outputOR	*Transition set as output XOR.*
cust.templateName	*Visual model template of diagram changed.*

Petri net arc property change events This type of event is related to changes of general and execution-specific properties of arcs in a Petri net diagram. All events of this type have a common prefix which is omitted in the following list, namely events.editor.petriNet.propertyChanged.properties.ArcsProperty.

Table B.11: Tracked events: Petri net arc property change events.

Event Type	Description
Multiplicator	*User changed the multiplier of the arc.*
Probability	*User changed the probability of activation in conflict resolution.*
ObjectCosts	*User changed the object passing costs of the arc.*
ObjectTimes	*User changed the object passing times of the arc.*
ServiceCondition	*User changed the BPEL service condition of the arc.*

Petri net classification property change events This type of event is related to changes of the status of Petri net diagrams and their importance for the phases of the Horus method. All events of this type have the common prefix events.editor.petriNet.propertyChanged.properties.Classification., which is omitted in the following list.

Table B.12: Tracked events: Petri net classification property change events.

Event Type	Description
ProgressStatus	*Progress status of the model changed.*
ProcStatus	*Comment about the progress status of the model changed.*
StrategyPhase	*Model association with the Horus method strategy phase changed.*
AnalysisPhase	*Model association with the Horus method analysis phase changed.*

Petri net cost property change events This type of event is related to changes of cost properties of the activities in a Petri net diagram. These properties allow specifying the cost that is incurred when an activity is executed as well as the expected added value. All events of this type have the common prefix events.editor.petriNet.propertyChanged.properties.Costs., which is omitted in the following list.

Table B.13: Tracked events: Petri net cost property change events.

Event Type	Description
Factor	*Cost factor which can be one, thousand, million, or billion.*
Processing.Type	*Type of indicated value (fixed or probability distribution).*
ProcessingMinimal	*Minimum processing cost.*
ProcessingAverage	*Average processing cost.*
ProcessingMaximal	*Maximum processing cost.*
Processing.Parameter	*Parameter possibly needed for probability distribution.*
Transport.Type	*Type of indicated value (fixed or probability distribution).*
TransportMinimal	*Minimum transport cost.*
TransportAverage	*Average transport cost.*
TransportMaximal	*Maximum transport cost.*
Transport.Parameter	*Parameter possibly needed for probability distribution.*
Added.Type	*Type of indicated value (fixed or probability distribution).*
MininmalAdded	*Minimum added value.*

Continued on the next page ▷

Table B.13 (contd.)

Event Type	Description
AverageAdded	*Average added value.*
MaximalAdded	*Maximum added value.*
Added.Parameter	*Parameter possibly needed for probability distribution.*

Petri net execution property change events This type of event is related to changes of execution properties of the activities in a Petri net diagram. These properties allow specifying the execution schedule and execution frequency of an activity. All events of this type have the common prefix events.editor.petriNet.propertyChanged.properties.Execution., which is omitted in the following list.

Table B.14: Tracked events: Petri net execution property change events.

Event Type	Description
TimeUnit	*Time unit of execution, ranging from millisecond to year.*
Number	*Execution frequency; how often the activity is executed.*
Monday	*Boolean indicating whether activity can be executed on Mondays.*
Tuesday	*Boolean indicating whether activity can be executed on Tuesdays.*
Wednesday	*Boolean indicating whether activity can be executed on Wednesdays.*
Thursday	*Boolean indicating whether activity can be executed on Thursdays.*
Friday	*Boolean indicating whether activity can be executed on Fridays.*
Saturday	*Boolean indicating whether activity can be executed on Saturdays.*
Sunday	*Boolean indicating whether activity can be executed on Sundays.*
StartTime	*Earliest time from which on activity execution is possible.*
StopTime	*Latest time until which activity execution is possible.*

Petri net general property change events This type of event is related to changes of general properties of a Petri net diagram and the Petri net model elements (i. e., excluding purely graphical elements such as rectangles or triangles) it contains. General properties are textual information that detail the semantics of the elements they are attached to. All events of this type have the common prefix events.editor.petriNet.propertyChanged.properties.General., which is omitted in the following list.

Table B.15: Tracked events: Petri net general property change events.

Event Type	Description
Diagram.LongName	*Long name of a Petri net diagram.*
Diagram.Description	*Detailed description of a Petri net diagram.*
Diagram.Notes	*Additional notes of a Petri net diagram.*
Node.Name	*Short name of a Petri net model element.*
Node.LongName	*Long name of a Petri net model element.*
Node.Description	*Detailed description of a Petri net model element.*
Node.Notes	*Additional notes for a Petri net model element.*

Petri net load generator property change events This type of event is related to load generation properties of activities in a Petri net diagram. The purpose of a load generator is to generate objects (i. e., tokens) for process simulation runs. All events of this type have a common prefix which the following list omits, i. e. events.editor.petriNet.propertyChanged.properties.loadGenerator.

Table B.16: Tracked events: Petri net load generator property change events.

Event Type	Description
enabled	*Transition/activity set/unset as load generator.*
repetitions	*Number of repetitions for load generation changed.*
webservice	*Web service to use for generation of objects changed.*
valuesFile	*File to use for generation of objects changed.*

Petri net object store property change events This type of event is related to changes of general properties of the object stores in a Petri net diagram. All events of this type have a common prefix which is omitted in the following list, namely events.editor.petriNet.propertyChanged.properties.ObjectStore.

Table B.17: Tracked events: Petri net object store property change events.

Event Type	Description
FetchStrategy	*Strategy for object consumption, e. g., First In First Out (FIFO).*
CostMultiplicator	*Cost multiplication factor for object storage.*
CostTimeUnit	*Time unit resp. interval for storage costs.*
MinimalCost	*Minimum storage costs.*
AverageCost	*Average storage costs.*
MaximalCost	*Maximum storage costs.*
QualityChange	*Quality change of stored objects in percent.*

Petri net rating property change events This type of event is related to changes of rating properties of the activities in a Petri net diagram. All events of this type have a common prefix which is omitted in the following list, namely events.editor.petriNet.propertyChanged.properties.Rating.

Table B.18: Tracked events: Petri net rating property change events.

Event Type	Description
KeyActivity	*Boolean: activity is a key activity.*
CostDriver	*Boolean: activity is a cost driver.*
ValueDriver	*Boolean: activity is a value driver.*
InnovationDriver	*Boolean: activity is an innovation driver.*
OptimizationPotential	*Boolean: activity has optimization potential.*
InternalAnnotations	*Internal annotations for the activity.*
Priority	*Priority of the activity as a number.*
TimelimitExceeding	*(Never fired.)*
Quality	*Impact of activity on the quality of produced objects.*
ErrorRate	*Error rate of activity in percent.*
Competitiveness	*Competitiveness of the activity as chosen from a list.*

Petri net simulation property change events This type of event is related to changes of simulation properties of the activities in a Petri net diagram. All events of this type have a common prefix which is omitted in the following list, namely events.editor.petriNet.propertyChanged.properties.simulation.

Table B.19: Tracked events: Petri net simulation property change events.

Event Type	Description
earliestStart	*Earliest start date of activity execution.*
latestStart	*Latest start date of activity execution.*
latestStop	*Latest end date of activity execution.*
multiExecution	*Boolean indicating whether multiple activity instances are possible.*
interruption	*Boolean indicating whether activity execution can be interrupted.*

Petri net time property change events This type of event is related to changes of temporal properties of the activities in a Petri net diagram. These properties allow specifying the required time for executing an activity. All events of this type have a common prefix which is omitted in the following list, namely events.editor.petriNet.propertyChanged.properties.Time.

Table B.20: Tracked events: Petri net time property change events.

Event Type	Description
Unit	*Time unit of all indicated times.*
Processing.Type	*Type of indicated time (fixed or probability distribution).*
ProcessingMinimal	*Minimum processing time.*
ProcessingAverage	*Average processing time.*
ProcessingMaximal	*Maximum processing time.*
Processing.Parameter	*Parameter possibly needed for probability distribution.*
Transport.Type	*Type of indicated time (fixed or probability distribution).*
TransportMinimal	*Minimum transport time.*
TransportAverage	*Average transport time.*
TransportMaximal	*Maximum transport time.*
Transport.Parameter	*Parameter possibly needed for probability distribution.*

Petri net Web service property change events This type of event is related to changes of service properties of the activities in a Petri net diagram. These properties are concerned with automated process execution and are specified using the BPEL as well as the Web Services Description Language (WSDL). All events of this type have a common prefix which the following list omits, i. e. events.editor.petriNet.propertyChanged.properties.WebService.

Table B.21: Tracked events: Petri net Web service property change events.

Event Type	Description
activity	*BPEL activity type, e. g., invoke or reply.*
wsdl	*URL of the WSDL file describing the service.*
partnerLinkType	*BPEL partner link type.*
portType	*BPEL port type.*
operation	*BPEL operation specifying an exchange of messages.*

Other Petri net property change events This type of event is related to properties of the elements in a Petri net diagram that cannot be assigned to any other category. All events of this type have the common prefix events.editor.petriNet.propertyChanged.properties., which is omitted in the following list.

Table B.22: Tracked events: Other Petri net property change events.

Event Type	Description
Implementation.Text	*Textual hints about the implementation of an element.*
Measures.measures	*List of measures related to a transition.*
reportingPriority	*Selection and order of elements to be included in a report.*

Petri net reference change events This type of event is related to the specification of references between the elements of a Petri net diagram and other diagrams or parts thereof. All events of this type have the common prefix events.editor.petriNet.propertyChanged.references., which is omitted in the following list.

Table B.23: Tracked events: Petri net reference change events.

Event Type	Description
entityType	*Reference from an object store to an object type.*
rule	*Reference from an activity to a business rule.*
role	*Reference from an activity to a role.*
kpi	*Reference from an activity to a KPI.*

Continued on the next page ▷

Table B.23 (contd.)

Event Type	Description
businessUnit	*Reference from an activity to an organizational unit.*
resource	*Reference from an activity to a resource.*
risk	*Reference from an activity to a risk.*
service	*Reference from an activity to a service.*
strategy	*(Never fired.)*
systemComponent	*Reference from an activity to a system component.*
refinement	*Reference from an activity to its refinement.*

B.2 Computed Measures

This section provides an exhaustive list of all measures that are currently computed by the gamification module of the Horus Business Modeler. It is subdivided according to the type of computation into standard measures, global measures, quality measures, and lastly calculated measures.

Standard Measures

Standard measures are quantitative indicators for the activity of a user conforming to the default case described in Section 7.2.2. The computation logic of standard measures is encapsulated by specific Java classes that can provide one or more measures. Furthermore, each standard measure defines one or more triggers, i. e., event types (cf. Section B.1) that cause the recomputation of its values. Triggers are not explicated in the following list, but can generally be derived from the textual description of each measure. Lastly, the values of standard measures are persisted in the database to facilitate their reuse. At the time of writing, the complete list of available standard measures is as follows:

measures.modeling.businessRules.numModelsCreated This measure gives a count of the number of business rule models created by a particular user.

measures.modeling.glossaries.numModelsCreated This measure counts the number of glossary models created by a particular user.

measures.modeling.objects.numModelsCreated This measure gives a count of the number of object models created by a particular user.

measures.modeling.employees.numModelsCreated This measure counts the number of employee models created by a particular user.

measures.modeling.processArchitectures.numModelsCreated This measure gives a count of the number of process architecture models created by a particular user.

measures.modeling.processes.numModelsCreated This measure counts the number of process models created by a particular user.

measures.modeling.contexts.numModelsCreated This measure gives a count of the number of context models created by a particular user.

measures.modeling.roles.numModelsCreated This measure gives a count of the number of role models created by a particular user.

measures.modeling.risks.numModelsCreated This measure gives a count of the number of risk models created by a particular user.

measures.modeling.goals.numModelsCreated This measure gives a count of the number of goal models created by a particular user.

measures.modeling.kpis.numModelsCreated This measure gives a count of the number of KPI models created by a particular user.

measures.modeling.swots.numModelsCreated This measure gives a count of the number of SWOT models created by a particular user.

measures.modeling.strategies.numModelsCreated This measure counts the number of strategy models created by a particular user.

measures.modeling.services.numModelsCreated This measure gives a count of the number of service models created by a particular user.

measures.modeling.businessUnits.numModelsCreated This measure gives a count of the number of business unit models created by a particular user.

measures.modeling.systemArchitectures.numModelsCreated This measure provides a count of the the number of system architecture models created by a particular user.

measures.modeling.organizationStructures.numModelsCreated This indicates the number of organization structure models created by a user.

measures.modeling.resources.numModelsCreated This measure counts the number of resource models created by a particular user.

measures.modeling.simulations.numModelsCreated This measure indicates the number of simulation models created by a particular user.

measures.modeling.templates.numModelsCreated This measure counts the number of template models created by a particular user.

measures.modeling.businessRules.numModelsDeleted This measure gives a count of the number of business rule models deleted by a particular user.

measures.modeling.glossaries.numModelsDeleted This measure counts the number of glossary models deleted by a particular user.

measures.modeling.objects.numModelsDeleted This measure gives a count of the number of object models deleted by a particular user.

measures.modeling.employees.numModelsDeleted This measure counts the number of employee models deleted by a particular user.

measures.modeling.processArchitectures.numModelsDeleted This measure gives a count of the number of process architecture models deleted by a particular user.

measures.modeling.processes.numModelsDeleted This measure counts the number of process models deleted by a particular user.

measures.modeling.contexts.numModelsDeleted This measure gives a count of the number of context models deleted by a particular user.

measures.modeling.roles.numModelsDeleted This measure gives a count of the number of role models deleted by a particular user.

measures.modeling.risks.numModelsDeleted This measure gives a count of the number of risk models deleted by a particular user.

measures.modeling.goals.numModelsDeleted This measure gives a count of the number of goal models deleted by a particular user.

measures.modeling.kpis.numModelsDeleted This measure gives a count of the number of KPI models deleted by a particular user.

measures.modeling.swots.numModelsDeleted This measure gives a count of the number of SWOT models deleted by a particular user.

measures.modeling.strategies.numModelsDeleted This measure counts the number of strategy models deleted by a particular user.

measures.modeling.services.numModelsDeleted This measure gives a count of the number of service models deleted by a particular user.

measures.modeling.businessUnits.numModelsDeleted This measure gives a count of the number of buiness unit models deleted by a particular user.

measures.modeling.systemArchitectures.numModelsDeleted This measure counts the number of system architecture models deleted by a particular user.

measures.modeling.organizationStructures.numModelsDeleted Counts the number of organization structure models deleted by a particular user.

measures.modeling.resources.numModelsDeleted This measure counts the number of resource models deleted by a particular user.

measures.modeling.simulations.numModelsDeleted This measure indicates the number of simulation models deleted by a particular user.

measures.modeling.templates.numModelsDeleted This measure counts the number of template models deleted by a particular user.

measures.user.login.numLoginsWeekend This measure gives a count of the number of times a particular user has logged into a repository on weekends (i. e., on Saturdays and Sundays).

measures.user.login.special.christmas2015 This measure gives a count of the number of times a particular user has logged into a repository on Christmas Eve 2015.

measures.user.login.special.christmas2016 This measure gives a count of the number of times a particular user has logged into a repository on Christmas Eve 2016.

measures.user.login.special.christmas2017 This measure gives a count of the number of times a particular user has logged into a repository on Christmas Eve 2017.

measures.user.login.special.newyear2016 This measure gives a count of the number of times a particular user has logged into a repository on New Year's Day 2016.

measures.user.login.special.newyear2017 This measure gives a count of the number of times a particular user has logged into a repository on New Year's Day 2017.

measures.user.gamification.points This measure gives a count of the total number of gamification-related points the user has obtained by creating high-quality process models or improving the quality of existing models.

measures.modeling.text.charactersWritten This measure gives a count of the total number of characters of text the user has written as textual documentation for the elements of process models. This includes the following fields: short name, name, description, and notes.

measures.modeling.pixelsPushed This measure gives a count of the total number of pixels that the user has displaced the elements of process models. Thus, a high value of this measure indicates that the user has invested time to rearrange model elements, possibly to improve readability.

measures.user.login.accountAgeDays This measure indicates the age of the account of a user in days.

measures.profile.completenessLevel This measure provides a summarized numerical indicator for the completeness level of a user's profile. To ensure that each possible value can be uniquely interpreted, powers of two are used for the following profile fields: first name (1), last name (2), description (4), email address (8). For example, a measure value of $9 = 8 + 1$ thus indicates that first name and email address have been specified.

measures.quality.model.perfectModels This measure counts the number of perfect models that a user has created. A model is considered to be perfect when all quality metrics that support it and that are activated for the current repository or workspace have a quality of 100% when the user triggers a save operation.

Global Measures

Global measures provide a single, global implementation that must be parameterized with a concrete event type (see Section B.1) when it is used. This has the main advantage that instead of requiring a separate implementation for each event type, a single implementation can perform the task for all of them. At the time of writing, the complete list of available global measures is as follows:

measures.count When parameterized with an arbitrary event type, this global measure provides a count of how often the user has caused this event to be fired in total. For instance, a value of 100 for events.user.login means that the user has logged in 100 times.

measures.largestGap When parameterized with an arbitrary event type, this global measure indicates the largest amount of time in days that has elapsed between two subsequent instances of the event type for a particular user. For instance, a value of 5 for events.user.login means that the longest duration during which the user has not logged in is 5 days.

measures.longestStreak When parameterized with an arbitrary event type, this global measure indicates the largest number of subsequent days during which at least one instance of the event type occurs on each day for a particular user. For instance, a value of 10 for events.user.login means that the largest number of subsequent days on which the user has logged in is 10.

Quality Measures

Quality measures are similar to global measures, but are parameterized with a quality algorithm rather than an event type. They are recomputed whenever the user saves a model and thus an event of the type events.quality.changed is fired. At the time of writing, the complete list of available quality measures is as follows:

measures.quality.metric.largestIncrease When parameterized with an arbitrary quality metric, this measure indicates the largest-ever increase of the value of the former through model changes made since the last save operation. For instance, a value of 10 for the metric edgeCrossings means that the largest number of edge crossings the user has ever added to a model is 10.

measures.quality.metric.largestDecrease When parameterized with an arbitrary quality metric, this measure indicates the largest-ever decrease of the value of the former through model changes made since the last save operation. For instance, a value of 10 for the metric edgeCrossings means that the largest number of edge crossings the user has ever removed from a model is 10.

measures.quality.metric.increaseSum When parameterized with an arbitrary quality metric, this measure indicates the total sum of value increases of the former through model changes. For instance, a value of 100 for the metric edgeCrossings means that the user has created a total number of 100 edge crossings.

measures.quality.metric.decreaseSum When parameterized with an arbitrary quality metric, this measure indicates the total sum of value decreases of the former through model changes. For instance, a value of 100 for the metric edgeCrossings means that the user has removed a total number of 100 edge crossings.

measures.quality.metric.changeSum When parameterized with an arbitrary quality metric, this measure indicates the sum of value changes of the former through model changes. This is analogous to the balance between value increases and decreases, and thus this measure is equivalent to the difference between the two measures increase sum and decrease sum. For instance, a value of -10 for the metric edgeCrossings means that the user has removed 10 more edge crossings than he has created.

measures.quality.metric.smallestValue When this measure is parameterized with an arbitrary quality metric, it indicates the smallest-ever value of the former the user has achieved through model changes. For instance, a value of 2 for edgeCrossings means that the user has never managed to create a model with less than 2 edge crossings.

measures.quality.metric.largestValue When parameterized with an arbitrary quality metric, this measure indicates the largest-ever value of the former the user has achieved through model changes. For instance, a value of 10 for edgeCrossings means that the user has never created a model with more than 10 edge crossings.

measures.quality.metric.highestQuality When parameterized with an arbitrary quality metric, this measure indicates the highest-ever quality level of the former the user has achieved through model changes. For instance, a value of 0.8 for the metric edgeCrossings means that the user has never managed to create a model with a better edge crossings quality than 80%.

measures.quality.metric.lowestQuality When this measure is parameterized with an arbitrary quality metric, it indicates the lowest-ever quality level of the former the user has achieved through model changes. For instance, the value 0.5 for the metric edgeCrossings means that the user has never created a model with a lower edge crossings quality than 50%.

Calculated Measures

Calculated measures are based on *expressions* with arbitrary nesting depth describing how their values should be computed. These expressions may consist of basic arithmetic operations (*addition, subtraction, multiplication, addition*), aggregation functions (*minimum, maximum, average*), *constant* values, and the values of other *measures* (including standard measures, global measures, quality measures, and other calculated measures). Calculated measures are not stored in the database and only computed when the value of an associated measure is updated. At the time of writing, the complete list of available calculated measures is as follows:

measures.modeling.numModelsCreated This measure gives a count of the number of models of any type created by a particular user. As such, it is calculated as the sum of all standard measures that conform to the following expression: measures.modeling.*.numModelsCreated.

measures.modeling.numModelsDeleted This measure gives a count of the number of models of any type deleted by a particular user. As such, it is calculated as the sum of all standard measures that conform to the following expression: measures.modeling.*.numModelsDeleted.

measures.profile.balanceSkillAdditionDeletion This measure provides the balance of skills added and deleted by a particular user to resp. from his profile. As such, it always indicates the number of skills that the user is currently claiming to possess. The value of this measure is computed as the difference between the global measure measures.count parameterized with the event type skill.added and the global measure measures.count parameterized with the event type skill.removed.

measures.profile.balanceSkillEndorsements This measure provides the balance of skills of other users endorsed and unendorsed by a particular user. As such, it always indicates the number of endorsements that the user is currently supporting. The value of this measure is computed as the difference between the global measure measures.count parameterized with the event type skill.endorsed and the global measure measures.count parameterized with the event type skill.unendorsed.

measures.profile.balanceContactAdditionDeletion This measure provides the balance of contact methods added and deleted by a particular user to resp. from his profile. As such, it always indicates the number of contact methods that the user is currently specifying. The value of this measure is computed as the difference between the global measure measures.count parameterized with the event type contact.added and the global measure measures.count parameterized with the event type contact.removed.

B.3 Measure Example

Listing B.1 illustrates the implementation of the standard measure measures.modeling.text.charactersWritten in the HBM. Due to the realization of a multi-layered class hierarchy for measures, most of the functionality of this class is implemented at higher levels (e. g., many methods are implemented in the class `AbstractMeasureImplementationDB`), and thus only the method `scoringFunction` must be implemented here. In this method, it is demonstrated how the XML use data contained in an event can be converted into Java objects to enable the computation of measure values. In this case, it is computed how many new characters of text a modeler has provided for a model element (e. g., in the form of a name, description, or note), and this difference is added to the previous value of the measure.

```
// Implementation of the measure CharactersWritten that indicates how many characters
// of textual documentation about model elements a user has written. Extends an
// abstract class with default implementations of many methods.
public class CharactersWritten extends AbstractMeasureImplementationDB {

    // Method that recomputes the measure based on its old value and a new event.
    @Override
    public Double scoringFunction(Double currentValue, HorusEventObject event) {
        try {
            // Step 1: Convert the Xml user data back into Java objects.
            PropertyChangeData changeData = event.convertData(PropertyChangeData.class,
            ↪  PropertyChangeData.XML_CLASSES);
            ValueObjectData oldValue = (ValueObjectData) changeData.getOldValue();
            ValueObjectData newValue = (ValueObjectData) changeData.getNewValue();

            // Step 2: Determine the number of new characters that have been written.
            // Note: Limited capability of detecting changes in the existing text.
            int oldChars = oldValue.getObject().toString().length();
            int newChars = newValue.getObject().toString().length();
            int charsWritten = Math.max(0, newChars - oldChars);

            // Step 3: Return the new value of the measure after this update.
            return currentValue + charsWritten;
        } catch (Exception ex) {
            return currentValue;
        }
    }
}
```

Listing B.1: Implementation of the measure CharactersWritten.

B.4 Quality Metric Example

Listing B.2 illustrates the implementation of the node overlaps metric in the HBM. In the method `getGoal`, the optimization goal for the quality metric is returned, which in this case is minimization. Furthermore, the method `computeBounds` sets values for the lower and upper bounds of the metric. Here, the lower bound is trivial, and the upper bound is set as the number of all nodes in the process model that occupy visual space. The actual computation of the metric is performed in the method `execute`. Specifically, this method computes the absolute measurement (i. e., the total number of node overlaps), which is converted into a relative measurement through the method `getQuality` situated in an abstract parent class. As it can be seen, the implementation makes no reference to a particular model type, and may thus be used for and kind of model for which an appropriate transformer class exists.

```java
// Implementation of the NodeOverlaps measure that indicates how many model elements
// overlap, thus reducing readability. The goal is to minimize this metric. The lower
// bound is 0 overlaps, the upper bound n overlaps with n = number of model elements.
public class NodeOverlaps extends AbstractReadabilityAlgorithm {

    // Method that indicates the optimization goal for the value of this metric.
    @Override
    public OptimizationGoal getGoal() { return OptimizationGoal.MINIMIZE; }

    // Method that determines the lower/upper bounds for the value of this metric.
    @Override
    public void computeBounds()
    {
        lowerBound = 0;
        upperBound = graph.getNodes().size();
    }

    // Method that computes the current value of this metric. Note that this is not
    // the quality itself, but the value that will be related to lower/upper bound.
    @Override
    public void execute() {
        // Step 1: Initialize based on assumption that there will be no overlaps.
        Map<QNode, Boolean> overlaps = new HashMap<QNode, Boolean>();
        currentValue = 0;

        // Step 2: Check for the trivial case that there is only one model element.
        if (graph.getNodes().size() <= 1) return;

        // Step 3: For each pair of nodes, check whether they visually overlap.
        for (int i = 0; i < graph.getNodes().size(); i++) {
            QNode outer = graph.getNodes().get(i);

            for (int j = i + 1; j < graph.getNodes().size(); j++) {
                QNode inner = graph.getNodes().get(j);

                // If a pair does overlap, remember the involved elements.
                if (QNode.testOverlap(outer, inner)) {
                    overlaps.put(outer, true);
                    overlaps.put(inner, true);
                }
            }
        }

        // Step 4: Set the new current value as the number of nodes having overlaps.
        currentValue = overlaps.size();
    }
}
```

Listing B.2: Implementation of the quality metric NodeOverlaps (some details left out).

C Field Experiment Details

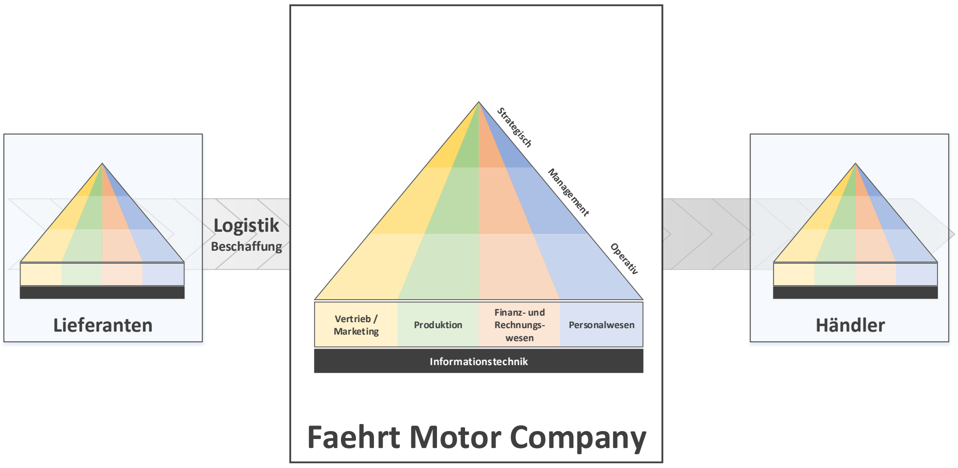
In this chapter, additional details about certain aspects of the field experiment that was conducted to evaluate the gamification module of the HBM are presented. Specifically, Section C.1 includes the full task description of the modeling case study students had to complete to successfully pass the lecture *Introduction to IS*. Furthermore, Section C.2 presents a small selection of slides that were used in the lecture in the winter term 2016/17 to communicate "soft goals" to the students. Then, Section C.3 presents the full results of the evaluation with regard to actual system use. The chapter ends with further details about the survey that was used to assess the user experience of students with the HBM in Section C.4.

C.1 Task Description

On the following pages, the full task description that was distributed to students at the beginning of the lecture *Introduction to IS* in all evaluated semesters is provided. While some details of this document changed over time, its key contents remained unchanged, and thus only the description used in the winter term 2014/15 is provided.

Einführung in die Wirtschaftsinformatik

Vorlesungsbegleitende Fallstudie, Wintersemester 2014/15



„Die *Faehrt Motor Company* mit Sitz in Delmenhorst (DE) ist nach Toyota, General Motors, Volkswagen und Hyundai der fünftgrößte Autohersteller weltweit (Stand 2011). Ursprung des Konzerns ist eine von Heinrich Faehrt in Dortmund 1903 gegründete Fabrik. Mit Einführung der Fließbandproduktion im Jahr 1913 brachte *Faehrt* einen radikalen Umbruch in der neu entstehenden Autoindustrie.“[1]

Es handelt sich bei diesem Unternehmen also um ein einerseits traditionelles Produktionsunternehmen mit großer Erfahrung im Bereich Prozessautomatisierung, andererseits um ein Unternehmen, das einem starken internationalen Innovations- und Wettbewerbsdruck ausgesetzt ist. Aus diesem Grund ist es von äußerster Wichtigkeit, dass alle internen und externen Prozesse sowie die interne und externe Kommunikation möglichst reibungslos und fehlerfrei durchgeführt werden. Unternehmen dieses Kalibers benötigen den Einsatz von Informationssystemen, um dies zu gewährleisten.

Sehen Sie sich während dieses Semesters als Mitarbeiter dieses internationalen Konzerns, das seinen Schwerpunkt im Bereich der Produktion von Fahrzeugen hat!

Innerhalb dieses Konzerns werden vier Abteilungen betrachtet:

- Produktion
- Vertrieb & Marketing
- Finanz- und Rechnungswesen
- Personalwesen

Daneben existieren noch Lieferanten, die die Faehrt Motor Company mit wichtigen Einzelteilen oder Modulen für den Autobau versorgen und der nicht zum Konzern gehörende Handel, welcher die fertigen Produkte schlussendlich an private und geschäftliche Abnehmer vertreibt. Ebenfalls im Rahmen der Fallstudie betrachtet wird die Beschaffungslogistik, die für die Lieferung von Materialien von Lieferanten an die Produktion verantwortlich ist.

[1] In (zufälliger) Anlehnung an http://de.wikipedia.org/wiki/Ford

Im Rahmen dieser Vorlesung werden Sie einer der vier internen Abteilungen oder einem der externen Partner Handel bzw. Beschaffungslogistik zugeordnet. Lieferanten werden nicht näher betrachtet. Gemeinsam mit Ihren Kollegen (das sind vier Ihrer Kommilitonen) werden Sie wöchentlich in Gruppenarbeit verschiedene Themengebiete und Methoden der Wirtschaftsinformatik, die Sie während der Vorlesung kennenlernen, auf den *Faehrt*-Case bzw. Ihre konkrete Abteilung bei *Faehrt* anwenden, Zusammenhänge diskutieren, Lösungen entwickeln und alles (!) dokumentieren.

Konkret bedeutet das für Sie:

- Im Rahmen der Veranstaltung erstellen Sie ein Handbuch für Ihre Abteilung. Dort dokumentieren Sie sämtliche Überlegungen und Lösungen, die Sie in Ihrer Gruppe während des Semesters erarbeiten.
- Dieses Handbuch gilt als Prüfungsleistung und ist am Ende des Semesters digital abzugeben.
- Sie erarbeiten zwei kurze Präsentationen (~10 Minuten), in denen Sie zunächst Ihren Zwischenstand (im Dezember) und dann Ihr Endergebnis (im Februar) darstellen.

Das Handbuch wird zum Teil mithilfe einer speziell auf die Bedürfnisse der Wirtschaftsinformatik ausgerichteten Software erstellt (für die Sie natürlich eine Schulung erhalten), zum anderen Teil mittels einer Textverarbeitung Ihrer Wahl. Im Handbuch wird Folgendes dokumentiert:

- Zu Beginn der Veranstaltung lernen Sie Ihre Gruppe kennen und entwickeln ein Verständnis für die Aufgaben Ihrer Abteilung (Was sind bspw. die Aufgaben der Produktionsabteilung? Welche Abläufe gibt es hier? Was sind die Schnittstellen zu den anderen Abteilungen?).
- Während des Semesters lernen Sie die unterschiedlichsten Sichtweisen der Wirtschaftsinformatik kennen und übertragen diese auf Ihre Abteilung (Was bedeutet bspw. IT-Sicherheit für die Personalabteilung? Wie ist bspw. die Produktion in die Supply Chain integriert?).
- Vereinzelt kann es auch konkrete Aufgaben zu den jeweiligen Veranstaltungen geben, die es zu bearbeiten gilt (Was ist bspw. Kerckhoffs' Prinzip? Wie wird ein „Prozess" definiert?).

Zur Diskussion Ihrer Fragen steht Ihnen im Learnweb ein Forum zur Verfügung, in dem Sie mit Ihren Kommilitonen Fragen offen diskutieren können und sollen. Die Betreuer werden – wenn nötig – in die Diskussion eingreifen.

C.2 Soft Goals

In the winter term 2016/17, the course participants were given additional hints and recommendations (i. e., soft goals) for when models can be considered "good enough" to be submitted for a passing grade. This supplemental information addressed the expected minimum model size, the need to write description texts for model elements and create references between process models, organizational diagrams, object models, and roles, and naming guidelines for model elements. Furthermore, students received explicit instructions to use the provided model quality feedback and the available help features. These requirements were communicated to students by means of particular lecture slides, which are depicted on the following pages.

RICHTLINIE 2

GEBT KEINE TRIVIALEN MODELLE AB

ERCIS

- Die Modelle, die ihr erstellt, müssen aussagekräftig sein
- Sie sollten daher eine vernünftige Größe haben und mehr Informationen mit höherer Detaillierung darstellen, als das Modell, für das die Verfeinerung erstellt wurde
- Eine pauschale Aussage zur minimalen Modellgröße lässt sich nicht machen; ein Modell mit weniger als vier Aktivitäten wird jedoch üblicherweise nicht ausreichen

RICHTLINIE 3

BENENNUNG VON MODELLEN UND MODELLELEMENTEN

ERCIS

- Prozessmodelle mit Level (Ebene) im Namen, bspw.:
  - L1 – Produktion
  - L2 – Bedarf prüfen
- Name soll zur Aktivität passen, die das Modell verfeinert
- Aktivitäten sollen mit <Nomen><Verb> benannt werden
  Bestellung verarbeiten, Rechnungseingang prüfen, …
- Objektspeicher sollen mit <Nomen> benannt sein
  Rechnung, Bestellung; oder auch “Eingehende Rechnung”

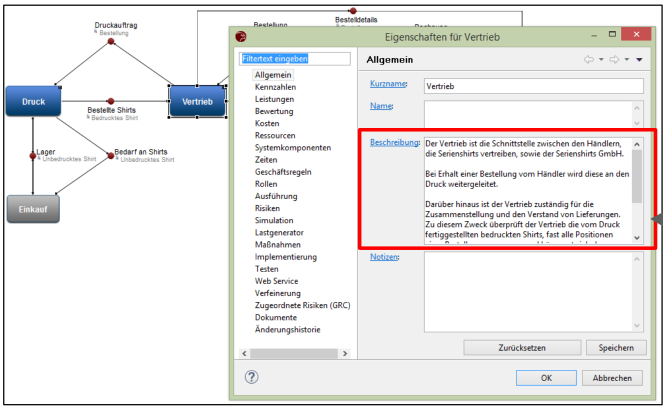
Horus Business Modeler: Richtlinien und Tipps
Nicolas Pflanzl, MSc
2
08.11.2016

RICHTLINIE 4

DOKUMENTATION

ERCIS

- Verseht all eure Aktivitäten, Objektspeicher, Organisationseinheiten, etc. mit **Beschreibungstexten**
- Diese scheinen im Report auf, den ihr am Ende abgebt



Je weniger selbsterklärend, desto länger sollte die Beschreibung sein

RICHTLINIE 5

VERKNÜPFUNG

ERCIS

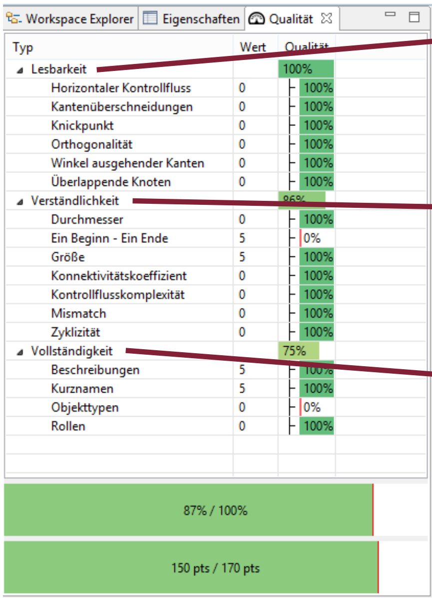
- Für die Fallstudie erstellt ihr nicht nur Prozessmodelle, sondern auch Objektmodelle, Rollen und Orga-Modelle

- **Vergesst nicht, diese miteinander zu verknüpfen:**
  - Rollen für Aktivitäten in Prozessmodellen
  - Objekttypen für Objektspeicher in Prozessmodellen
  - Rollen für Organisationseinheiten in Orga-Modellen
  - Organisationseinheiten für Objekte in Objektmodellen

Horus Business Modeler: Richtlinien und Tipps
Nicolas Pflanzl, MSc
4
08.11.2016

TIPP 2

NUTZT DAS QUALITÄTSFEEDBACK

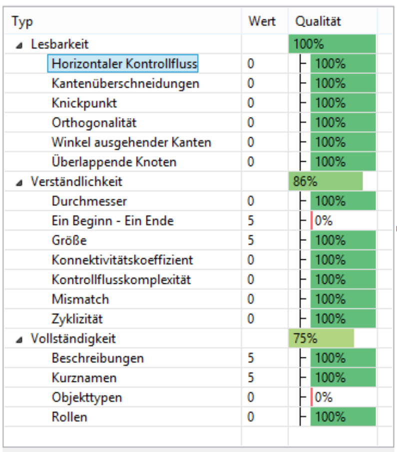


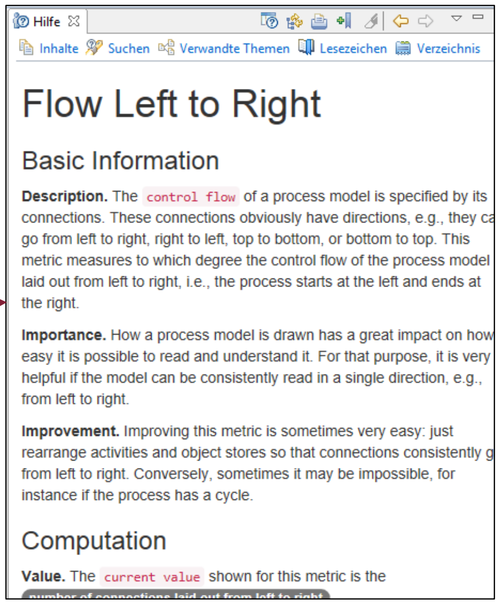
- **Ist euer Modell gut lesbar?**
  Bezieht sich auf das Layout
- **Ist das Modell gut verständlich?**
  Bezieht sich auf Modellinhalt
- **Ist das Modell vollständig?**
  Bezieht sich auf Modellinhalt
  Sollte in Abgabe 100% sein!

TIPP 2

NUTZT DAS QUALITÄTSFEEDBACK

- Nähere Infos zu einer Metrik: anklicken und F1 drücken





Horus Business Modeler: Richtlinien und Tipps
Nicolas Pflanzl, MSc
6
08.11.2016

C.3 System Use

This section provides additional result data regarding system use. Specifically, Table C.2 and Table C.1 are extended versions of Table 8.19 and Table 8.20 without omitted lines.

Table C.1: Field experiment: Descriptive statistics for system use (full).

	Gamification			**~~Gamification~~**			
Event Type	**Mean**	**SD**	**Med.**	**Mean**	**SD**	**Med.**	**Diff.**
events	1597.76	948.34	1473	1496.05	990.54	1211	▲0.18
editor	1527.19	921.87	1417	1462.07	975.08	1191	▲0.16
modelClosed	57.75	35.81	49	41.37	23.82	34	▲0.31
modelOpened	61.77	36.87	56	44.14	24.44	36	▲0.36
modelSaved	34.34	20.9	29	27.72	17.81	26	▲0.1
petriNet	1373.33	861.96	1294	1348.8	922.96	1100	▲0.15
created	357.75	228.95	322	380.81	295.62	310	▲0.04
graphics	4.28	14.78	0	6.3	23.71	0	=
ellipsis	0.01	0.11	0	0	0	0	=
freeLine	1.13	5.63	0	0.58	2.51	0	=
freeShape	0.11	0.6	0	0	0	0	=
freeText	1.92	9.23	0	4.6	23.3	0	=
image	0.06	0.4	0	0	0	0	=
note	0.47	3.13	0	0.56	1.69	0	=
rectangle	0.51	2.92	0	0.56	1.82	0	=
rectangleRound	0.06	0.56	0	0.47	1.78	0	=
model	353.47	225.35	322	374.51	289.33	308	▲0.04
arc	193.04	124.05	174	218.12	197.07	157	▲0.1
place	101.13	64.29	94	97.3	68.82	82	▲0.13
transition	59.3	39.96	55	59.09	42.93	54	▲0.02
deleted	9.58	14.67	5	11.95	18.95	4	▲0.2
graphics	0.42	1.94	0	0.37	1.29	0	=
freeLine	0.1	0.79	0	0.05	0.21	0	=
freeShape	0.03	0.23	0	0	0	0	=
freeText	0.15	0.91	0	0.28	1.22	0	=
note	0.08	0.5	0	0	0	0	=
rectangle	0.06	0.56	0	0.05	0.21	0	=
rectangleRound	0.06	0.56	0	0.02	0.15	0	=
model	9.16	14.05	4	11.58	18.66	4	=
arc	6.16	9.36	3	7.6	14.33	3	=

Continued on the next page ▷

Table C.1 (contd.)

Event Type	Gamification Mean	SD	Med.	~~Gamification~~ Mean	SD	Med.	Diff.
place	1.67	2.97	0	2.56	3.64	1	▼1
transition	1.33	2.41	0	1.42	2.46	0	=
propertyChanged	1006	633.32	936	956.07	643.56	787	▲0.16
graphics	706.14	450.17	618	693.58	474.15	565	▲0.09
addVirtTokens	0.43	3.38	0	0.12	0.76	0	=
alphaWert	0.38	2.19	0	0	0	0	=
arrangeNodes	1.06	2.13	0	0.53	1.1	0	=
backgroundColor	19	13.7	17	20.93	15.18	20	▼0.15
bendpoint	0.04	0.19	0	0	0	0	=
bordering	36.18	24.63	32	41.09	29	40	▼0.2
capacity	79.08	55.99	69	77.07	58.36	67	▲0.03
cust	3.67	4.01	3	3.63	5.09	2	▲0.33
arcType	0.57	1.37	0	0.58	1.85	0	=
inAndOutputOR	0.22	0.61	0	0.14	0.41	0	=
inputOR	0.48	1.23	0	0.58	1.24	0	=
outputOR	2.41	2.68	2	2.33	3.54	1	▲0.5
defaultImage	133.35	89.14	122	125.3	86.19	111	▲0.09
diagramName	0.18	0.64	0	0.09	0.29	0	=
enabled	0.65	3.89	0	0.35	1.46	0	=
endArrowFill	0.08	0.57	0	0	0	0	=
endArrowStyle	0.06	0.56	0	0	0	0	=
endArrowX	1.03	9.11	0	0.02	0.15	0	=
endArrowY	1.03	9.11	0	0	0	0	=
fill	0.04	0.19	0	0.05	0.21	0	=
firing	0.43	3.38	0	0.12	0.76	0	=
foregroundColor	0.67	2.6	0	0.37	1.2	0	=
gradient	0.44	1.45	0	0.56	1.44	0	=
gradientColor	0.42	1.46	0	0.42	1.16	0	=
gridColor	0.01	0.11	0	0.44	2.35	0	=
gridEnabled	0.04	0.34	0	0	0	0	=
gridSize	0.24	1.23	0	0.37	1.23	0	=
gridVisible	0.08	0.68	0	0	0	0	=
horizontalAlign	0	0	0	0.21	1.37	0	=
image	0.48	1.07	0	0.35	0.75	0	=
imageLabels	0.01	0.11	0	0.05	0.21	0	=
lineStyle	16.59	23.07	10	16.58	22.47	8	▲0.2
lineWidth	150.29	106.98	127	142.09	101.58	126	▲0.01

Continued on the next page ▷

Table C.1 (contd.)

Event Type	**Gamification** Mean	SD	**Med.**	**~~Gamification~~** Mean	SD	**Med.**	**Diff.**
location	202.44	140.33	173	189.67	130.2	172	▲0.01
pointList	1.52	8.22	0	0.74	3.5	0	=
removeVirtTokens	0.34	3.04	0	0.12	0.76	0	=
scale	0.04	0.25	0	0	0	0	=
size	46.23	32.2	39	58.7	57.89	36	▲0.08
subdiagram	4.78	4.45	3	5.56	4.15	5	▼0.4
text	2.34	9.33	0	5.19	22.52	0	=
tokenCount	0.29	2.27	0	0.93	6.1	0	=
virtTokenCount	1.39	7.86	0	1.63	10.67	0	=
xorConnection	1.15	2.98	0	0.65	1.67	0	=
properties	284.46	186.81	246	253.49	180.67	204	▲0.17
Classification	0.01	0.11	0	0	0	0	=
ProcStatus	0.01	0.11	0	0	0	0	=
Costs	0.05	0.45	0	0	0	0	=
ProcessingMax	0.03	0.23	0	0	0	0	=
ProcessingMin	0.01	0.11	0	0	0	0	=
TransportMax	0.01	0.11	0	0	0	0	=
General	241.68	157.34	224	209.05	151.47	176	▲0.21
Diagram	0.27	0.76	0	0.35	0.97	0	=
Description	0.27	0.76	0	0.35	0.97	0	=
Node	241.42	157.44	224	208.7	151.38	176	▲0.21
Description	72.37	60.84	67	48.95	63.97	30	▲0.55
LongName	6	20.65	0	7.7	33.04	0	=
Name	162.76	107.22	155	151.77	97.93	144	▲0.07
Notes	0.29	1.03	0	0.28	1.83	0	=
Time	0.15	1.35	0	0	0	0	=
ProcessingMax	0.06	0.56	0	0	0	0	=
ProcessingMin	0.06	0.56	0	0	0	0	=
TransportMax	0.01	0.11	0	0	0	0	=
TransportMin	0.01	0.11	0	0	0	0	=
WebService	41.61	32.86	38	42.63	33.62	35	▲0.08
activity	40.54	32.1	37	41.42	33.15	35	▲0.05
partnerLinkType	0.27	0.93	0	0.3	1.28	0	=
portType	0.27	0.93	0	0.3	1.28	0	=
wsdl	0.53	1.86	0	0.6	2.56	0	=
reportingPrio	0.7	3.19	0	1.53	6.18	0	=
simulation	0.25	0.95	0	0.28	1.1	0	=

Continued on the next page ▷

Table C.1 (contd.)

Event Type	Gamification Mean	SD	Med.	~~Gamification~~ Mean	SD	Med.	Diff.
earliestStart	0.08	0.31	0	0.09	0.37	0	=
latestStart	0.09	0.33	0	0.09	0.37	0	=
latestStop	0.09	0.33	0	0.09	0.37	0	=
references	15.41	14.55	13	9	12.42	5	▲0.62
entityType	7.27	9.67	2	4.05	7.48	0	▲1
resource	0.78	2.54	0	0	0	0	=
role	7.27	7.58	5	4.91	6.87	2	▲0.6
rule	0.09	0.29	0	0.05	0.3	0	=
help	1.08	2.74	0	0.58	1.31	0	=
read	1.08	2.74	0	0.58	1.31	0	=
achievements	0.11	0.42	0	0.07	0.34	0	=
businessProcesses	0.11	0.32	0	0.09	0.29	0	=
gamiUse	0.13	0.33	0	0.05	0.21	0	=
gamiUseOrga	0.11	0.36	0	0.02	0.15	0	=
gamification	0.11	0.32	0	0.09	0.29	0	=
petriNets	0.08	0.27	0	0.12	0.32	0	=
profileEdit	0.27	0.69	0	0.07	0.26	0	=
rewards	0.1	0.3	0	0.07	0.26	0	=
uploadPhoto	0.06	0.33	0	0	0	0	=
projectManager	18.24	8.23	17	15.84	7.75	14	▲0.18
modelCreated	12.91	4.98	13	11.72	5.39	11	▲0.15
modelCopied	2.81	7.36	0	1.81	4.68	0	=
modelDeleted	0.01	0.11	0	0.07	0.26	0	=
modelRenamed	2.71	3.4	2	2.49	3.03	1	▲0.5
workspaceCreated	0.1	0.38	0	0.05	0.21	0	=
workspaceCopied	0.08	0.35	0	0.02	0.15	0	=
workspaceRenamed	0.01	0.11	0	0	0	0	=
quality	25.75	19.09	22	4.88	14.32	0	▲1
changed	25.75	19.09	22	4.88	14.32	0	▲1
reward	14.46	8.74	13	4.09	3.96	3	▲0.77
badge	4.11	1.55	4	3.02	1.34	3	▲0.25
unlocked	4.11	1.55	4	3.02	1.34	3	▲0.25
points	10.34	7.98	8	1.07	3.2	0	▲1
received	10.34	7.98	8	1.07	3.2	0	▲1
user	11.05	5.86	10	8.58	4.34	8	▲0.2
login	10.56	5.7	10	8.3	4.32	7	▲0.3
pic	0.06	0.33	0	0.02	0.15	0	=

Continued on the next page ▷

Table C.1 (contd.)

Event Type	Gamification Mean	SD	Med.	~~Gamification~~ Mean	SD	Med.	Diff.
uploaded	0.06	0.33	0	0.02	0.15	0	=
profile	0.43	0.52	0	0.26	0.44	0	=
changed	0.43	0.52	0	0.26	0.44	0	=
Event Types	55	8.12	54	49.83	7.17	49	▲0.09
Session Length	-	-	-	-	-	-	-
Sum	106868	81762	96042	97989	102478	63245	▲0.34
Avg w/ Empty	11974	12115	8538	12919	19653	7177	▲0.16
Avg w/o Empty	14093	13019	10174	15939	21007	9854	▲0.03
Avg. Decision Time	84.52	96.29	50.78	77.23	83.3	50.84	=

Table C.2: Field experiment: System use hypothesis test results (full).

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
events	✔	1556.0	-0.7637	0.2237	-	-	-
editor	✔	1589.5	-0.5841	0.2807	-	-	-
modelClosed	✖	1210.0	-2.6183	0.0043	***	0.2370	M
modelOpened	✖	1183.0	-2.7630	0.0027	***	0.2501	M
modelSaved	✔	1362.0	-1.8043	0.0356	**	0.1634	S
petriNet	✔	1635.0	-0.3403	0.3678	-	-	-
created	✔	1697.5	-0.0054	0.4984	-	-	-
graphics	✔	1519.0	-1.2172	0.1139	-	-	-
ellipsis	✔	1677.0	-0.7378	0.6475	-	-	-
freeLine	✔	1641.0	-0.5568	0.3016	-	-	-
freeShape	✖	1634.0	-1.2884	0.2678	-	-	-
freeText	✔	1598.5	-0.7722	0.2173	-	-	-
image	✖	1655.5	-1.0477	0.4174	-	-	-
note	✔	1569.0	-1.6175	0.0929	*	0.1464	S
rectangle	✔	1609.0	-1.0083	0.1766	-	-	-
rectangleRound	✖	1600.5	-1.7024	0.0695	*	0.1541	S
model	✔	1689.0	-0.0509	0.4803	-	-	-
arc	✖	1691.5	-0.0375	0.4856	-	-	-
place	✔	1597.5	-0.5413	0.2953	-	-	-
transition	✔	1669.5	-0.1554	0.4390	-	-	-
deleted	✔	1636.0	-0.3378	0.3686	-	-	-
graphics	✔	1614.5	-0.9463	0.2316	-	-	-

Continued on the next page ▷

Table C.2 (contd.)

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
freeLine	✔	1663.5	-0.6080	0.4418	-	-	-
freeShape	✔	1677.0	-0.7378	0.6475	-	-	-
freeText	✔	1644.5	-0.7723	0.2834	-	-	-
note	✖	1655.5	-1.0477	0.4174	-	-	-
rectangle	✔	1642.0	-1.1287	0.2834	-	-	-
rectangleRound	✔	1681.0	-0.4264	0.5826	-	-	-
model	✔	1605.0	-0.5061	0.3073	-	-	-
arc	✔	1680.0	-0.1010	0.4609	-	-	-
place	✔	1459.5	-1.3924	0.0830	*	0.1261	S
transition	✔	1697.5	-0.0059	0.4962	-	-	-
propertyChanged	✔	1594.5	-0.5573	0.2898	-	-	-
graphics	✔	1630.5	-0.3644	0.3588	-	-	-
addVirtTokens	✔	1632.0	-0.9512	0.2686	-	-	-
alphaWert	✖	1634.0	-1.2884	0.2678	-	-	-
arrangeNodes	✔	1462.5	-1.4531	0.0724	*	0.1316	S
backgroundColor	✔	1590.5	-0.5791	0.2824	-	-	-
bendpoint	✖	1634.0	-1.2885	0.2678	-	-	-
bordering	✔	1547.5	-0.8095	0.2103	-	-	-
capacity	✔	1602.0	-0.5172	0.3037	-	-	-
cust	✔	1508.5	-1.0343	0.1514	-	-	-
arcType	✔	1616.5	-0.6335	0.2801	-	-	-
inAndOutputOR	✔	1654.0	-0.4070	0.3363	-	-	-
inputOR	✔	1682.0	-0.1187	0.4488	-	-	-
outputOR	✔	1519.0	-0.9911	0.1618	-	-	-
defaultImage	✔	1602.0	-0.5172	0.3037	-	-	-
diagramName	✔	1678.5	-0.2076	0.3929	-	-	-
enabled	✔	1666.5	-0.3203	0.4252	-	-	-
endArrowFill	✔	1655.5	-1.0477	0.4174	-	-	-
endArrowStyle	✔	1677.0	-0.7378	0.6475	-	-	-
endArrowX	✔	1681.0	-0.4264	0.5826	-	-	-
endArrowY	✔	1677.0	-0.7378	0.6475	-	-	-
fill	✔	1684.0	-0.2263	0.5796	-	-	-
firing	✔	1632.0	-0.9512	0.2686	-	-	-
foregroundColor	✔	1662.0	-0.3335	0.4000	-	-	-
gradient	✔	1618.0	-0.7355	0.2474	-	-	-
gradientColor	✔	1639.5	-0.5714	0.3254	-	-	-
gridColor	✖	1640.0	-1.1686	0.1223	-	-	-
gridEnabled	✔	1677.0	-0.7378	0.6475	-	-	-

Continued on the next page ▷

Table C.2 (contd.)

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
gridSize	✔	1589.0	-1.2948	0.1334	-	-	-
gridVisible	✔	1677.0	-0.7378	0.6475	-	-	-
horizontalAlign	✖	1659.0	-1.3554	0.3525	-	-	-
image	✔	1635.0	-0.4469	0.3302	-	-	-
imageLabels	✖	1641.0	-1.1487	0.2834	-	-	-
lineStyle	✔	1660.0	-0.2085	0.4185	-	-	-
lineWidth	✔	1607.5	-0.4877	0.3140	-	-	-
location	✔	1600.0	-0.5279	0.2999	-	-	-
pointList	✔	1620.0	-0.7375	0.2431	-	-	-
removeVirtTokens	✔	1681.0	-0.4264	0.5826	-	-	-
scale	✖	1655.5	-1.0477	0.4174	-	-	-
size	✖	1670.5	-0.1501	0.4411	-	-	-
subdiagram	✔	1452.5	-1.3290	0.0926	*	0.1203	S
text	✔	1397.0	-2.1938	0.0160	**	0.1986	S
tokenCount	✔	1696.0	-0.0499	0.7166	-	-	-
virtTokenCount	✔	1654.0	-0.6943	0.4204	-	-	-
xorConnection	✔	1627.0	-0.5204	0.3007	-	-	-
properties	✔	1497.0	-1.0799	0.1410	-	-	-
Classification	✔	1677.0	-0.7378	0.6475	-	-	-
ProcStatus	✔	1677.0	-0.7378	0.6475	-	-	-
Costs	✔	1677.0	-0.7378	0.6475	-	-	-
ProcessingMax	✔	1677.0	-0.7378	0.6475	-	-	-
ProcessingMin	✔	1677.0	-0.7378	0.6475	-	-	-
TransportMax	✔	1677.0	-0.7378	0.6475	-	-	-
General	✔	1452.0	-1.3210	0.0939	*	0.1196	S
Diagram	✔	1669.5	-0.2650	0.3655	-	-	-
Description	✔	1669.5	-0.2650	0.3655	-	-	-
Node	✔	1450.0	-1.3318	0.0921	*	0.1206	S
Description	✔	1196.0	-2.6974	0.0033	***	0.2442	M
LongName	✔	1630.5	-0.4883	0.3209	-	-	-
Name	✔	1616.0	-0.4421	0.3303	-	-	-
Notes	✔	1549.0	-1.6843	0.0747	*	0.1525	S
Time	✔	1677.0	-0.7378	0.6475	-	-	-
ProcessingMax	✔	1677.0	-0.7378	0.6475	-	-	-
ProcessingMin	✔	1677.0	-0.7378	0.6475	-	-	-
TransportMax	✔	1677.0	-0.7378	0.6475	-	-	-
TransportMin	✔	1677.0	-0.7378	0.6475	-	-	-
WebService	✔	1689.5	-0.0482	0.4814	-	-	-

Continued on the next page ▷

Table C.2 (contd.)

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
activity	✔	1671.0	-0.1474	0.4422	-	-	-
partnerLinkType	✔	1666.0	-0.3253	0.4223	-	-	-
portType	✔	1666.0	-0.3253	0.4223	-	-	-
wsdl	✔	1666.0	-0.3253	0.4223	-	-	-
reportingPrio	✖	1664.0	-0.4587	0.2767	-	-	-
simulation	✔	1690.5	-0.0946	0.5813	-	-	-
earliestStart	✔	1686.5	-0.1499	0.4796	-	-	-
latestStart	✔	1689.5	-0.1065	0.5813	-	-	-
latestStop	✔	1689.5	-0.1065	0.5813	-	-	-
references	✔	1116.5	-3.1310	0.0008	***	0.2835	M
entityType	✖	1295.5	-2.2730	0.0110	**	0.2058	M
resource	✖	1419.0	-2.7970	0.0024	***	0.2532	M
role	✔	1269.0	-2.3291	0.0097	***	0.2109	M
rule	✔	1591.0	-1.3433	0.1565	-	-	-
help	✖	1581.5	-0.7954	0.2144	-	-	-
read	✖	1581.5	-0.7954	0.2144	-	-	-
achievements	✔	1629.0	-0.8222	0.2758	-	-	-
businessProcesses	✔	1663.0	-0.3560	0.4899	-	-	-
gamiUse	✖	1562.5	-1.4130	0.1341	-	-	-
gamiUseOrga	✖	1565.5	-1.5738	0.0986	*	0.1425	S
gamification	✔	1663.0	-0.3560	0.4899	-	-	-
petriNets	✔	1630.0	-0.7400	0.3326	-	-	-
profileEdit	✖	1530.0	-1.5410	0.0567	*	0.1395	S
rewards	✔	1645.0	-0.5779	0.4125	-	-	-
uploadPhoto	✖	1634.0	-1.2885	0.2678	-	-	S
projectManager	✔	1314.5	-2.0609	0.0195	**	0.1866	S
modelCreated	✔	1373.5	-1.7463	0.0405	**	0.1581	S
modelCopied	✔	1547.5	-1.0349	0.1526	-	-	-
modelDeleted	✖	1601.5	-1.6853	0.1250	-	-	-
modelRenamed	✔	1617.5	-0.4423	0.3305	-	-	-
workspaceCreated	✔	1646.5	-0.6497	0.2962	-	-	-
workspaceCopied	✔	1651.0	-0.7411	0.2957	-	-	-
workspaceRenamed	✔	1677.0	-0.7378	0.6475	-	-	-
quality	✖	389.5	-7.1526	0.0000	***	0.6476	L
changed	✖	389.5	-7.1526	0.0000	***	0.6476	L
reward	✖	355.0	-7.2167	0.0000	***	0.6534	L
badge	✔	958.0	-4.0423	0.0000	***	0.3660	L
unlocked	✔	958.0	-4.0423	0.0000	***	0.3660	L

Continued on the next page ▷

Table C.2 (contd.)

Event Type	Hom.	U-Value	Z-Value	p-Value	Sig.	Effect	Size
points	✖	313.0	-7.5865	0.0000	***	0.6868	L
received	✖	313.0	-7.5865	0.0000	***	0.6868	L
user	✔	1276.0	-2.2688	0.0115	**	0.2054	M
login	✔	1305.5	-2.1110	0.0173	**	0.1911	S
pic	✔	1672.5	-0.4517	0.3646	-	-	-
uploaded	✔	1672.5	-0.4517	0.3646	-	-	-
profile	✖	1418.0	-1.8032	0.0474	**	0.1633	S
changed	✖	1418.0	-1.8032	0.0474	*	0.1633	S
Event Types	✔	1085.0	-3.2921	0.0004	***	0.2980	M
Session Length							
Sum	✔	1414.0	-1.3339	0.0919	*	0.1208	S
Avg w/ Empty	✖	1598.0	-0.5386	0.2967	-	-	-
Avg w/o Empty	✖	1634.0	-0.3457	0.3663	-	-	-
Avg. Decision Time	✔	1589.0	-0.5868	0.2803	-	-	-

C.4 Measurement Instrument

In Table C.3, detailed information about the survey used in the field experiment is provided. Specifically, the first column contains the textual definition of each question, the second column depicts their internal question codes, the third column assigns them to a particular construct that was measured, and the last column describes the scale on which each question was measured.

Table C.3: Field experiment: User experience survey details.

Question	Code	Construct	Scale
Horus lässt sich einfach benutzen	U1	Usability	7-point Likert
Die Funktionen von Horus sind genau richtig für meine Ziele	N1	Usefulness	7-point Likert
Das Design von Horus wirkt attraktiv	A1	Aesthetics	7-point Likert
Es wird schnell klar, wie man Horus bedienen muss	U2	Usability	7-point Likert
Ich halte Horus für absolut nützlich	N2	Usefulness	7-point Likert
Die Bedienung von Horus ist verständlich	U3	Usability	7-point Likert
Mit Horus kann ich meine Ziele erreichen	N3	Usefulness	7-point Likert
Horus nervt mich	EN1	Neg. Emotions	7-point Likert
Durch Horus fühle ich mich erschöpft	EN2	Neg. Emotions	7-point Likert
Durch Horus fühle ich mich ausgeglichen	EP1	Pos. Emotions	7-point Likert
Horus frustriert mich	EN3	Neg. Emotions	7-point Likert
Horus stimmt mich euphorisch	EP2	Pos. Emotions	7-point Likert
Durch Horus fühle ich mich passiv	EN4	Neg. Emotions	7-point Likert
Horus beruhigt mich	EP3	Pos. Emotions	7-point Likert
Durch Horus fühle ich mich fröhlich	EP4	Pos. Emotions	7-point Likert
Horus verärgert mich	EN5	Neg. Emotions	7-point Likert
Ich möchte Horus auch in anderen Vorlesungen nutzen	NI1	Usage Intention	7-point Likert
Ich würde Horus gegen kein anderes Produkt eintauschen	L1	Loyalty	7-point Likert
Ich würde Horus auch im privaten/geschäftlichen Umfeld nutzen	NI2	Usage Intention	7-point Likert
Müsste ich EWI noch einmal belegen, würde ich erneut Horus nutzen wollen	L2	Loyalty	7-point Likert
Wenn ich mit Horus zu tun habe, vergesse ich schon mal die Zeit	NI3	Usage Intention	7-point Likert
Geben Sie an, wie Sie den Horus Business Modeler insgesamt bewerten	G1	Overall Grade	School grade

D Laboratory Experiment Details

In this chapter, additional details about certain aspects of the laboratory experiment that was conducted to evaluate the gamification module of the HBM are presented. Specifically, Section D.1 includes the full survey that was used in the experiment, and Section D.2 illustrates how its individual question items are mapped to their respective constructs.

D.1 Task Description

This section contains the full measurement instrument (i. e., questionnaire) that was distributed to the participants of the laboratory experiment. The first page briefly explains the experiment to students as an assessment of process modeling skills and contains some items related to demographic information. On the second page, participants are provided with a short explanation about accessing the Horus server on which the modeling task was carried out. The description of the task itself, i. e., the textual representation of the process to model, is contained on the third page. Finally, the remaining four pages contain question items related to flow, intrinsic motivation, extrinsic motivation, technology acceptance, and process modeling self-efficacy.

Assessment von Modellierungsfähigkeiten

Vielen Dank für deine Teilnahme an diesem Assessment!

Zur Verbesserung der Lehre möchten wir herausfinden, wie gut du durch die aktuelle Form der Vorlesung *Einführung in die Wirtschaftsinformatik* **Prozessmodellierung erlernt** hast. Darüber hinaus hilft uns deine Teilnahme bei aktuellen Forschungsarbeiten.

Das Assessment besteht aus einer Modellierungsaufgabe im Schwierigkeitsgrad einer Klausur und mehreren Fragebögen. Die Teilnahme sollte insgesamt **ca. 45 Minuten** in Anspruch nehmen.

Die Befragung ist **vollständig anonym**! Über die vierstellige Kennung `0739` wirst du später die Möglichkeit haben, deine eigenen Ergebnisse einzusehen. Für andere Teilnehmer sind sie nicht sichtbar.

Bitte mache vor Beginn der Aufgabe ein paar Angaben zu deiner Person.

Demographische Informationen

Alter: ☐ Jahre

Geschlecht: ☐ Männlich ☐ Weiblich

Wie viele Stunden spielst du in der Woche durchschnittlich Computer- und Videospiele? ☐ Stunden

Vorwissen

Seit wann modellierst du Prozesse? ☐ Semester

Wie schätzt du dich selbst als Prozessmodellierer ein?

0	10	20	30	40	50	60	70	80	90	100
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Anfänger					*Fortgeschritten*					*Experte*

Vorbereitung

Hast du dich auf dieses Assessment vorbereitet? ☐ Ja ☐ Nein

Falls ja, wie?

____

____

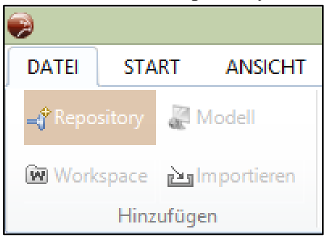
____

- 0739 -

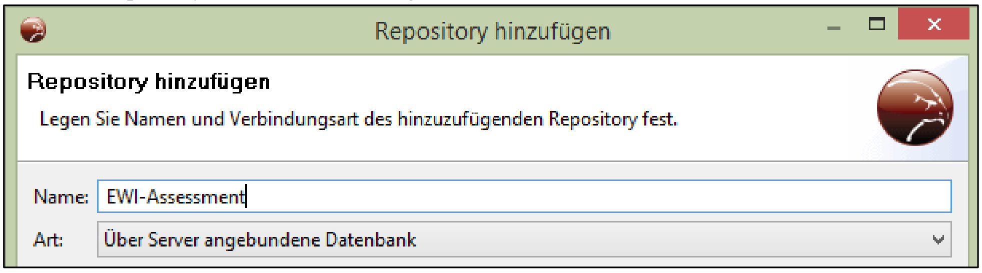
Bitte behalte diese Seite nach Bearbeitung der Aufgabe, damit du zu einem späteren Zeitpunkt über deine Kennung deine Ergebnisse nachschlagen kannst!

Einrichtung von Horus

Schritt 1: Neues Repository erstellen.



Schritt 2: Repository-Namen und –Art eingeben.



Schritt 3: Verbindungsdaten eingeben.

Webservice-URL:	`https://horus.biz/unimuensteval/horus/`
Benutzername:	`0739`
Passwort:	`rzj2`

Schritt 4: Eigenen Workspace identifizieren.

Der Name deines Workspace ist identisch zu deinem Benutzernamen: 0739

Schritt 5: Modellierungsaufgabe beginnen.

Siehe nächste Seite.

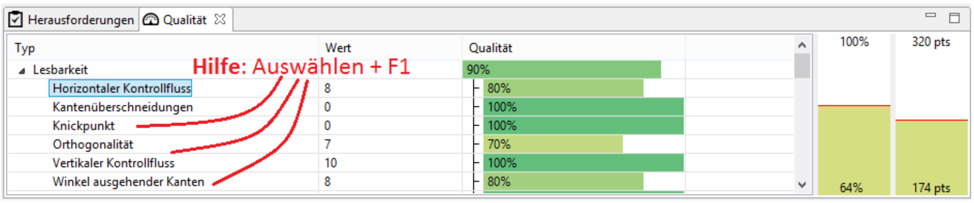
- 0739 -

Aufgabe

Du bist Mitarbeiter der Unternehmensberatung „Mc Allan" und hast von der Stadt Osnabrück den Auftrag bekommen, zur späteren Verbesserung die Prozesse im Bürgerservice zu analysieren. Ein Mitarbeiter schildert dir einen Teil des Prozesses zur **Ausstellung des neuen Personalausweises**:

> *Seit 2010 haben die Bürger in Osnabrück die Möglichkeit, neben dem herkömmlichen Personalausweis, den es mit oder ohne eine Signaturkomponente gibt, den Personalausweis mit biometrischen Sicherheitsmerkmalen zu erhalten. Um diesen zu beantragen spricht der Antragsteller in der Abteilung „Bürgerservice" im Bürgeramt vor. Hat der Antragsteller nicht alle notwendigen Unterlagen dabei, wird er wieder nach Hause geschickt; ansonsten wird mit der Datenerfassung begonnen. Dabei legt der Sachbearbeitet das Foto des Antragstellers in ein elektrisches Zuschneidegerät ein, welches dieses automatisch in das passende Format schneidet. Gleichzeitig erfasst der Sachbearbeiter die Personaldaten des Antragstellers in der Software MESO, wofür sich dieser durch ein bestehendes Ausweisdokument (bspw. alter Ausweis, Reisepass, Geburtsurkunde) authentifizieren muss.*
>
> *Im Anschluss daran fragt der Sachbearbeiter, ob die Nutzung der neuen biometrischen Merkmale gewünscht ist. Wird dies vom Antragsteller gewünscht, werden zwei Fingerabdrücke genommen; hierfür nutzt er das Softwaremodul MESO-FP. Die Fingerabdrücke werden durch das Modul sofort auf ihre Qualität hin überprüft. Sollte ihre Lesbarkeit nicht gegeben sein, wird die Erfassung wiederholt. In beiden Fällen druckt der Sachbearbeiter abschließend eine Übersicht über die Personaldaten und klebt das Passbild auf den einen Ausdruck. Der Antragsteller begleicht daraufhin den Unkostenbeitrag und die fertigen Antragsdaten werden zur Bundesdruckerei gesendet.*

Bitte erstelle aus dem Bericht ein Prozessmodell als Petri-Netz. Erfasse dabei, wo aus der Beschreibung ersichtlich, relevante Rollen und Ressourcen wie Geräte und Anwendungssysteme. Versuche, ein Modell mit möglichst hoher <u>Qualität</u> zu erzeugen. Das Modell sollte so lesbar, verständlich und vollständig sein, wie möglich. Nutze dafür die in Horus integrierten Hilfen in den Tabs „Qualität" und „Herausforderung". Über F1 erhältst du weitere Informationen zu einer einzelnen Qualitätsmetrik. Beachte, dass es nicht immer möglich ist, eine Qualität von 100% zu erreichen.



- 0739 -

ZWISCHENBEFRAGUNG

Bitte unterbrich kurz die Bearbeitung der Aufgabe, um zu einigen Aussagen Stellung zu beziehen. Die Bedeutung der Antwortoptionen lautet dabei wie folgt:

1	2	3	4	5	6	7
Trifft überhaupt nicht zu	Trifft größtenteils nicht zu	Trifft eher nicht zu	Weder noch	Trifft eher zu	Trifft größtenteils zu	Trifft voll und ganz zu

	1	2	3	4	5	6	7
Meine Gedanken bzw. Aktivitäten verlaufen flüssig und glatt.	☐	☐	☐	☐	☐	☐	☐
Ich bin ganz vertieft in das, was ich gerade mache.	☐	☐	☐	☐	☐	☐	☐
Es steht etwas für mich Wichtiges auf dem Spiel.	☐	☐	☐	☐	☐	☐	☐
Mein Kopf ist völlig klar.	☐	☐	☐	☐	☐	☐	☐
Ich mache mir Sorgen über einen Misserfolg.	☐	☐	☐	☐	☐	☐	☐
Die richtigen Gedanken kommen wie von selbst.	☐	☐	☐	☐	☐	☐	☐
Ich darf jetzt keine Fehler machen.	☐	☐	☐	☐	☐	☐	☐
Ich bin völlig selbstvergessen	☐	☐	☐	☐	☐	☐	☐
Ich fühle mich optimal beansprucht.	☐	☐	☐	☐	☐	☐	☐
Ich merke gar nicht, wie die Zeit vergeht.	☐	☐	☐	☐	☐	☐	☐
Ich habe das Gefühl, den Ablauf unter Kontrolle zu haben.	☐	☐	☐	☐	☐	☐	☐
Ich weiß bei jedem Schritt, was ich zu tun habe.	☐	☐	☐	☐	☐	☐	☐
Ich habe keine Mühe, mich zu konzentrieren.	☐	☐	☐	☐	☐	☐	☐

Verglichen mit allen anderen Tätigkeiten, die ich sonst mache, ist die jetzige Tätigkeit …

Leicht								Schwer
☐	☐	☐	☐	☐	☐	☐	☐	☐

Ich denke meine Fähigkeiten auf diesem Gebiet sind …

Niedrig								Hoch
☐	☐	☐	☐	☐	☐	☐	☐	☐

Für mich persönlich sind die jetzigen Anforderungen …

Zu gering				Gerade richtig				Zu hoch
☐	☐	☐	☐	☐	☐	☐	☐	☐

- 0739 -

Schlussbefragung – Teil 1

Bitte nimm zum Abschluss noch zu einigen Aussagen zu Horus und zu deinen Empfindungen beim Bearbeiten der Aufgabe Stellung. Die Bedeutung der Antwortoptionen lautet dabei wie folgt:

1	2	3	4	5	6	7
Trifft überhaupt nicht zu	Trifft größtenteils nicht zu	Trifft eher nicht zu	Weder noch	Trifft eher zu	Trifft größtenteils zu	Trifft voll und ganz zu

	1	2	3	4	5	6	7
Ich würde die Aufgabe als sehr unterhaltsam bezeichnen.	☐	☐	☐	☐	☐	☐	☐
Während ich an der Aufgabe gearbeitet habe, dachte ich daran, wie sehr sie mir gefällt.	☐	☐	☐	☐	☐	☐	☐
Ich finde Horus für die Prozessmodellierung nützlich.	☐	☐	☐	☐	☐	☐	☐
Ich fühlte mich während der Aufgabe unter Druck gesetzt.	☐	☐	☐	☐	☐	☐	☐
Die Qualität meiner mit Horus erstellten Prozessmodelle ist hoch.	☐	☐	☐	☐	☐	☐	☐
Ich habe mir bei dieser Aufgabe große Mühe gegeben.	☐	☐	☐	☐	☐	☐	☐
Es hat mir sehr gefallen, die Aufgabe zu machen.	☐	☐	☐	☐	☐	☐	☐
Ich bewerte die Qualität meiner mit Horus erstellten Prozessmodelle als exzellent.	☐	☐	☐	☐	☐	☐	☐
Ich habe diese Aufgabe für die Bezahlung gemacht.	☐	☐	☐	☐	☐	☐	☐
Ich habe an der Qualität meiner mit Horus erstellten Prozessmodelle nichts zu beanstanden.	☐	☐	☐	☐	☐	☐	☐
Ohne die Bezahlung hätte ich diese Aufgabe nicht gemacht.	☐	☐	☐	☐	☐	☐	☐
Ich dachte, dass die Aufgabe sehr langweilig war.	☐	☐	☐	☐	☐	☐	☐
Bei der Prozessmodellierung Horus zu nutzen erhöht meine Produktivität.	☐	☐	☐	☐	☐	☐	☐
Ich bin zufrieden mit meiner Leistung bei dieser Aufgabe.	☐	☐	☐	☐	☐	☐	☐
Ich plane, Horus im weiteren Verlauf meines Studiums zu nutzen.	☐	☐	☐	☐	☐	☐	☐
Ich fühlte mich entspannt, während ich die Aufgabe gemacht habe.	☐	☐	☐	☐	☐	☐	☐
Ich habe mich bei dieser Aufgabe ziemlich fähig gefühlt.	☐	☐	☐	☐	☐	☐	☐
Ich habe diese Aufgabe gemacht, um mir zu zeigen, dass ich eine Klausur bestanden hätte.	☐	☐	☐	☐	☐	☐	☐
Ich dachte die Aufgabe war sehr interessant.	☐	☐	☐	☐	☐	☐	☐

- 0739 -

1	2	3	4	5	6	7
Trifft überhaupt nicht zu	Trifft größtenteils nicht zu	Trifft eher nicht zu	Weder noch	Trifft eher zu	Trifft größtenteils zu	Trifft voll und ganz zu

	1	2	3	4	5	6	7
Meine Interaktion mit Horus ist klar und verständlich.	☐	☐	☐	☐	☐	☐	☐
Ich war unruhig, während ich die Aufgabe gemacht habe.	☐	☐	☐	☐	☐	☐	☐
Horus zu nutzen verbessert meine Leistung bei der Prozessmodellierung.	☐	☐	☐	☐	☐	☐	☐
Die Interaktion mit Horus erfordert nicht viel geistige Anstrengung.	☐	☐	☐	☐	☐	☐	☐
Ich war überhaupt nicht nervös davor, die Aufgabe anzugehen.	☐	☐	☐	☐	☐	☐	☐
Ich finde Horus einfach zu verwenden.	☐	☐	☐	☐	☐	☐	☐
Es war wichtig für mich, bei dieser Aufgabe gut abzuschneiden.	☐	☐	☐	☐	☐	☐	☐
Ich habe nicht viel Energie in diese Aufgabe gesteckt.	☐	☐	☐	☐	☐	☐	☐
Ich fand die Aufgabe sehr interessant.	☐	☐	☐	☐	☐	☐	☐
Ich finde es einfach, mit Horus das zu tun, was ich tun will.	☐	☐	☐	☐	☐	☐	☐
Nachdem ich eine Weile an der Aufgabe gearbeitet hatte, fühlte ich mich ziemlich kompetent.	☐	☐	☐	☐	☐	☐	☐
Ich habe mir nicht viel Mühe gegeben, bei dieser Aufgabe gut abzuschneiden.	☐	☐	☐	☐	☐	☐	☐
Ich fühlte mich angespannt, während ich die Aufgabe gemacht habe.	☐	☐	☐	☐	☐	☐	☐
Ich sage voraus, dass ich Horus wieder nutzen würde, wenn sich eine Gelegenheit ergäbe.	☐	☐	☐	☐	☐	☐	☐
Ich plane Horus zu nutzen, wenn sich eine Gelegenheit ergibt.	☐	☐	☐	☐	☐	☐	☐
Ich habe viel Aufwand in diese Aufgabe gesteckt.	☐	☐	☐	☐	☐	☐	☐
Ich habe diese Aufgabe gemacht, um mir zu beweisen, dass ich gut genug bin.	☐	☐	☐	☐	☐	☐	☐
Horus für die Prozessmodellierung zu nutzen verbessert meine Effektivität.	☐	☐	☐	☐	☐	☐	☐
Ich denke ich bin in dieser Aufgabe ziemlich gut.	☐	☐	☐	☐	☐	☐	☐
Ich denke ich habe mich bei dieser Aktivität im Vergleich zu anderen Studenten ziemlich gut geschlagen.	☐	☐	☐	☐	☐	☐	☐
Die Aufgabe zu machen hat Spaß gemacht.	☐	☐	☐	☐	☐	☐	☐

- 0739 -

Schlussbefragung – Teil 2

Die Modellierung von Geschäftsprozessen ist eine herausfordernde Aufgabe. Bitte gib im Folgenden deine Zuversicht in diesem Moment an, diese Herausforderungen bei der Bearbeitung einer anderen Modellierungsaufgabe mit gleicher Schwierigkeit zu überwinden.

Bitte bewerte deine Zuversicht anhand folgender Skala mit einer Zahl zwischen 0 und 100:

0	10	20	30	40	50	60	70	80	90	100
Keine Zuversicht					Moderate Zuversicht					Hohe Zuversicht

	Zuversicht (0-100)
Aktivitäten/Objekte in einer textuellen Beschreibung eines Prozesses identifizieren.	___________
Aus einer textuellen Beschreibung eines Prozesses ein korrektes Modell erstellen.	___________
Sequenzielle Aktivitäten in einem Prozessmodell korrekt wiedergeben.	___________
Parallele Aktivitäten in einem Prozessmodell korrekt wiedergeben.	___________
Alternative Aktivitäten in einem Prozessmodell korrekt wiedergeben.	___________
Iterative Aktivitäten in einem Prozessmodell korrekt wiedergeben.	___________
Ein qualitativ hochwertiges Prozessmodell erstellen.	___________
Ein Prozessmodell mit hoher Lesbarkeit erstellen.	___________
Ein Prozessmodell mit hoher Verständlichkeit erstellen.	___________
Ein Prozessmodell mit hoher Vollständigkeit erstellen.	___________

- 0739 -

D.2 Measurement Instrument

Table D.1 provides a different view on the survey included in the previous section. Specifically, it lists all question items and demonstrates to which measured constructs they belong. To that extent, the first column of the table depicts the position of each item in the questionnaire, with D1-D7 describing the positions of items relating to demographic information, and Q1-Q66 indicating the positions of all questions related to motivational constructs and technology acceptance. In the second column, the full textual representation of each question is provided. Furthermore, the third column marks all those questions with the letter "R" whose answer scores should be reversed when calculating construct scores. These questions are formulated in a "reverse" (often negative) fashion to reduce answer bias. The internal codes that refer to each individual question are depicted in the fourth column. Finally, the last column indicates on which scale each question was measured. Here, the following values are possible: INT (Integer number), M/F (gender, male or female), 0-100, BIN (binary choice, yes or no), TXT (free text), LIK7 (7-point Likert scale), and LIK9 (9-point Likert scale).

Table D.1: Laboratory experiment: Survey details.

Pos.	Question	R	Code	Scale
Demographic Information				
D1	*Alter*		D1	INT
D2	*Geschlecht*		D2	M/F
D3	*Wie viele Stunden spielst du in der Woche durchschnittlich Computer- und Videospiele?*		D3	INT
D4	*Seit wann modellierst du Prozesse?*		D4	INT
D5	*Wie schätzt du dich selbst als Prozessmodellierer ein?*		D5	0-100
D6	*Hast du dich auf dieses Assessment vorbereitet?*		D7	BIN
D7	*Falls ja, wie?*		D8	TXT
Flow: Smooth Progression				
Q1	*Meine Gedanken bzw. Aktivitäten verlaufen flüssig und glatt.*		FG1	LIK7
Q13	*Ich habe keine Mühe, mich zu konzentrieren.*		FG2	LIK7
Q4	*Mein Kopf ist völlig klar.*		FG3	LIK7
Q6	*Die richtigen Gedanken kommen wie von selbst.*		FG4	LIK7
Q12	*Ich weiß bei jedem Schritt, was ich zu tun habe.*		FG5	LIK7

Continued on the next page ▷

Table D.1 (contd.)

Pos.	Question	R	Code	Scale
Q11	*Ich habe das Gefühl, den Ablauf unter Kontrolle zu haben.*		FG6	LIK7
Flow: Absorbedness				
Q9	*Ich fühle mich optimal beansprucht.*		FA1	LIK7
Q10	*Ich merke gar nicht, wie die Zeit vergeht.*		FA2	LIK7
Q2	*Ich bin ganz vertieft in das, was ich gerade mache.*		FA3	LIK7
Q8	*Ich bin völlig selbstvergessen.*		FA4	LIK7
Flow: Anxiety				
Q3	*Es steht etwas für mich Wichtiges auf dem Spiel.*		FB1	LIK7
Q7	*Ich darf jetzt keine Fehler machen.*		FB2	LIK7
Q5	*Ich mache mir Sorgen über einen Misserfolg.*		FB3	LIK7
Flow: Skill-Challenge Fit				
Q14	*Verglichen mit allen anderen Tätigkeiten, die ich sonst mache, ist die jetzige Tätigkeit …*		FP1	LIK9
Q15	*Ich denke meine Fähigkeiten auf diesem Gebiet sind …*		FP2	LIK9
Q16	*Für mich persönlich sind die jetzigen Anforderungen …*		FP3	LIK9
Intrinsic Motivation: Interest/Enjoyment				
Q18	*Während ich an der Aufgabe gearbeitet habe, dachte ich daran, wie sehr sie mir gefällt.*		II1	LIK7
Q44	*Ich fand die Aufgabe sehr interessant.*		II2	LIK7
Q56	*Die Aufgabe zu machen hat Spaß gemacht.*		II3	LIK7
Q23	*Es hat mir sehr gefallen, die Aufgabe zu machen.*		II4	LIK7
Q28	*Ich dachte, dass die Aufgabe sehr langweilig war.*	R	II5	LIK7
Q35	*Ich dachte die Aufgabe war sehr interessant.*		II6	LIK7
Q17	*Ich würde die Aufgabe als sehr unterhaltsam bezeichnen.*		II7	LIK7
Intrinsic Motivation: Perceived Competence				
Q54	*Ich denke ich bin in dieser Aufgabe ziemlich gut.*		IC1	LIK7
Q55	*Ich denke ich habe mich bei dieser Aktivität im Vergleich zu anderen Studenten ziemlich gut geschlagen.*		IC2	LIK7
Q30	*Ich bin zufrieden mit meiner Leistung bei dieser Aufgabe.*		IC3	LIK7
Q33	*Ich habe mich bei dieser Aufgabe ziemlich fähig gefühlt.*		IC4	LIK7
Q46	*Nachdem ich eine Weile an der Aufgabe gearbeitet hatte, fühlte ich mich ziemlich kompetent.*		IC5	LIK7
Intrinsic Motivation: Pressure/Tension				
Q40	*Ich war überhaupt nicht nervös davor, die Aufgabe anzugehen.*	R	IP1	LIK7

Continued on the next page ▷

Table D.1 (contd.)

Pos.	Question	R	Code	Scale
Q48	*Ich fühlte mich angespannt, während ich die Aufgabe gemacht habe.*		IP2	LIK7
Q32	*Ich fühlte mich entspannt, während ich die Aufgabe gemacht habe.*	R	IP3	LIK7
Q37	*Ich war unruhig, während ich die Aufgabe gemacht habe.*		IP4	LIK7
Q20	*Ich fühlte mich während der Aufgabe unter Druck gesetzt.*		IP5	LIK7
Intrinsic Motivation: Effort/Importance				
Q51	*Ich habe viel Aufwand in diese Aufgabe gesteckt.*		IE1	LIK7
Q47	*Ich habe mir nicht viel Mühe gegeben, bei dieser Aufgabe gut abzuschneiden.*	R	IE2	LIK7
Q22	*Ich habe mir bei dieser Aufgabe große Mühe gegeben.*		IE3	LIK7
Q42	*Es war wichtig für mich, bei dieser Aufgabe gut abzuschneiden.*		IE4	LIK7
Q43	*Ich habe nicht viel Energie in diese Aufgabe gesteckt.*	R	IE5	LIK7
Extrinsic Motivation: External Regulation				
Q25	*Ich habe diese Aufgabe für die Bezahlung gemacht.*		ER1	LIK7
Q27	*Ohne die Bezahlung hätte ich diese Aufgabe nicht gemacht.*		ER2	LIK7
Extrinsic Motivation: Introjected Regulation				
Q34	*Ich habe diese Aufgabe gemacht, um mir zu zeigen, dass ich eine Klausur bestanden hätte.*		IR1	LIK7
Q52	*Ich habe diese Aufgabe gemacht, um mir zu beweisen, dass ich gut genug bin.*		IR2	LIK7
Technology Acceptance: Perceived Usefulness				
Q38	*Horus zu nutzen verbessert meine Leistung bei der Prozessmodellierung.*		PU1	LIK7
Q29	*Bei der Prozessmodellierung Horus zu nutzen erhöht meine Produktivität.*		PU2	LIK7
Q53	*Horus für die Prozessmodellierung zu nutzen verbessert meine Effektivität.*		PU3	LIK7
Q19	*Ich finde Horus für die Prozessmodellierung nützlich.*		PU4	LIK7
Technology Acceptance: Perceived Ease of Use				
Q36	*Meine Interaktion mit Horus ist klar und verständlich*		PEOU1	LIK7
Q39	*Die Interaktion mit Horus erfordert nicht viel geistige Anstrengung.*		PEOU2	LIK7
Q41	*Ich finde Horus einfach zu verwenden.*		PEOU3	LIK7
Q45	*Ich finde es einfach, mit Horus das zu tun, was ich tun will.*		PEOU4	LIK7

Continued on the next page ▷

Table D.1 (contd.)

Pos.	Question	R	Code	Scale
Technology Acceptance: Output Quality				
Q21	*Die Qualität meiner mit Horus erstellten Prozessmodelle ist hoch.*		OUT1	LIK7
Q26	*Ich habe an der Qualität meiner mit Horus erstellten Prozessmodelle nichts zu beanstanden.*		OUT2	LIK7
Q24	*Ich bewerte die Qualität meiner mit Horus erstellten Prozessmodelle als exzellent.*		OUT3	LIK7
Technology Acceptance: Behavioral Intention				
Q50	*Ich plane Horus zu nutzen, wenn sich eine Gelegenheit ergibt.*		BI1	LIK7
Q49	*Ich sage voraus, dass ich Horus wieder nutzen würde, wenn sich eine Gelegenheit ergäbe.*		BI2	LIK7
Q31	*Ich plane, Horus im weiteren Verlauf meines Studiums zu nutzen.*		BI3	LIK7
Process Modeling Self-Efficacy				
Q57	*Akt./Obj. in einer textuellen Beschreibung eines Prozesses identifizieren.*		PMSE1	0-100
Q58	*Aus textueller Beschr. eines Prozesses ein korrektes Modell erstellen.*		PMSE2	0-100
Q59	*Sequenzielle Aktivitäten in einem Prozessmodell korrekt wiedergeben.*		PMSE3	0-100
Q60	*Parallele Aktivitäten in einem Prozessmodell korrekt wiedergeben.*		PMSE4	0-100
Q61	*Alternative Aktivitäten in einem Prozessmodell korrekt wiedergeben.*		PMSE5	0-100
Q62	*Iterative Aktivitäten in einem Prozessmodell korrekt wiedergeben.*		PMSE6	0-100
Q63	*Ein qualitativ hochwertiges Prozessmodell erstellen.*		PMSE7	0-100
Q64	*Ein Prozessmodell mit hoher Lesbarkeit erstellen.*		PMSE8	0-100
Q65	*Ein Prozessmodell mit hoher Verständlichkeit erstellen.*		PMSE9	0-100
Q66	*Ein Prozessmodell mit hoher Vollständigkeit erstellen.*		PMSE10	0-100

Bibliography

[AAAM97] Gustavo Alonso, Divyakant Agrawal, Amr E. Abbadi, and Carl Mohan. Functionality and Limitations of Current Workflow Management Systems. *IEEE Expert Intelligent Systems & Their Applications*, 12(5):1–25, 1997.

[AB01] Craig A. Anderson and Brad J. Bushman. Effects of Violent Video Games on Aggressive Behavior, Aggressive Cognition, Aggressive Affect, Physiological Arousal, and Prosocial Behavior: A Meta-Analytic Review of the Scientific Literature. *Psychological Science*, 12(5):353–360, 2001.

[ABS11] Evmorfia N. Argyriou, Michael A. Bekos, and Antonios Symvonis. Maximizing the Total Resolution of Graphs. In *Proceedings of the 18th International Symposium on Graph Drawing (GD 2010)*, pages 62–67, Konstanz, Germany, 2011.

[Abt70] Clark C. Abt. *Serious Games*. Viking Press, New York, NY, United States, 1st edition, 1970.

[AC11] Judd Antin and Elizabeth F. Churchill. Badges in Social Media: A Social Psychological Perspective. In *Proceedings of the CHI 2011 Workshop on Gamification: Using Game Design Elements in NonGame Contexts*, Vancouver, BC, Canada, 2011.

[ACD+10] Patrizio Angelini, Luca Cittadini, Giuseppe Di Battista, Walter Didimo, Fabrizio Frati, Michael Kaufmann, and Antonios Symvonis. On the Perspectives Opened by Right Angle Crossing Drawings. In *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*, pages 21–32, Chicago, IL, United States, 2010.

[ACUV16] Dominik Augenstein, Murat Citak, Meike Ullrich, and Arthur Vetter. Experience Report: Social BPM Lab Enhanced with Participation of Professionals. In *Proceedings of the Workshop on Modellierung in der Hochschullehre at the Modellierung 2016 Conference*, pages 71–78, Karlsruhe, Germany, 2016.

[AD12] Ernest Adams and Joris Dormans. *Game Mechanics: Advanced Game Design*. New Riders Games, Berkeley, CA, United States, 1st edition, 2012.

[ADL13] Ahmed Awad, Gero Decker, and Niels Lohmann. ISEAsy: A Social Business Process Management Platform. In *Proceedings of the 6th Workshop on Business*

Process Management and Social Software (BPMS 2013), pages 125–137, Beijing, China, 2013.

[AEHO14] Sascha Alpers, Esmahan Eryilmaz, Stefan Hellfeld, and Andreas Oberweis. Mobile Modeling Tool Based on the Horus Method. In *Proceedings of the 3rd International Workshop on Advanced Information Systems for Enterprises (IWAISE 2014)*, pages 65–71, Tunis, Tunisia, 2014.

[AF80] Icek Ajzen and Martin Fishbein. *Understanding Attitudes and Predicting Social Behavior.* Prentice Hall, Upper Saddle River, NJ, United States, 1st edition, 1980.

[AF08] Thilini R. Ariyachandra and Mark N. Frolick. Critical Success Factors in Business Performance Management—Striving for Success. *Information Systems Management*, 25(2):113–120, 2008.

[AGS92] D. E. Avison, P. A. Golder, and H. U. Shah. Towards an SSM Toolkit: Rich Picture Diagramming. *European Journal of Information Systems*, 1(6):397–407, 1992.

[AHKL13] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering User Behavior with Badges. In *Proceedings of the 22nd International Conference on World Wide Web (WWW 2013)*, pages 95–106, Rio de Janeiro, Brazil, 2013.

[AIS$^+$77] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Angel Shlomo. *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press, Oxford, United Kingdom, 1st edition, 1977.

[And04] Craig A. Anderson. An Update on the Effects of Playing Violent Video Games. *Journal of Adolescence*, 27(1):113–122, 2004.

[AS04] Ruth Sara Aguilar-Savén. Business Process Modelling: Review and Framework. *International Journal of Production Economics*, 90(2):129–149, 2004.

[AS07] Toshiki Ami and Rainer Sommer. Comparison and Evaluation of Business Process Modelling and Management Tools. *International Journal of Services and Standards*, 3(2):249–261, 2007.

[ASI$^+$10] Craig A. Anderson, Akiko Shibuya, Nobuko Ihori, Edward L. Swing, Brad J. Bushman, Akira Sakamoto, Hannah R. Rothstein, and Muniba Saleem. Violent Video Game Effects on Aggression, Empathy, and Prosocial Behavior in Eastern and Western Countries: A Meta-Analytic Review. *Psychological Bulletin*, 136(2):151–173, 2010.

[AVSM12] Andrés Francisco Aparicio, Francisco Luis Gutiérrez Vela, José Luis González Sánchez, and José Luis Isla Montes. Analysis and Application of Gamification. In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador (INTERACCION 2012)*, Elche, Spain, 2012.

[BA02] Åsa Blomquist and Mattias Arvola. Personas in Action: Ethnography in an Interaction Design Team. In *Proceedings of the 2nd Nordic Conference on Human-Computer Interaction (NORDCHI 2002)*, pages 197–200, Aarhus, Denmark, 2002.

[BA03] Robert Blumberg and Shaku Atre. The Problem with Unstructured Data. *DM Review*, 13:42–46, 2003.

[Bag07] Richard P. Bagozzi. The Legacy of the Technology Acceptance Model and a Proposal for a Paradigm Shift. *Journal of the Association for Information Systems*, 8(4):244–254, 2007.

[Ban77] Albert Bandura. Toward a Unifying Theory of Behavioral Change. *Psychological Review*, 84(2):191–215, 1977.

[Ban82] Albert Bandura. Self-Efficacy Mechanism in Human Agency. *American Psychologist*, 37(2):122–147, 1982.

[Ban06] Albert Bandura. Guide for Constructing Self-Efficacy Scales. In *Self-Efficacy Beliefs of Adolescents*, chapter 14, pages 307–337. Information Age Publishing, Charlotte, NC, United States, 1st edition, 2006.

[Bar96] Richard A. Bartle. Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs. *Journal of MUD Research*, 1(1):1–19, 1996.

[Bat04] Bob Bates. *Game Design*. Thomson Course Technology, Boston, MA, United States, 2nd edition, 2004.

[BAZN13] Jose Berengueres, Fatma Alsuwairi, Nazar Zaki, and Tony Ng. Gamification of a Recycle Bin with Emoticons. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2013)*, pages 83–84, Tokyo, Japan, 2013.

[BBFRS12] Cynthis Beath, Irma Becerra-Fernandez, Jeanne Ross, and James Short. Finding Value in the Information Explosion. *MIT Sloan Management Review*, 53(4):18–20, 2012.

[BCC+10] Wasana Bandara, Donald R. Chand, Alina M. Chircu, Sandra Hintringer, Dimitris Karagiannis, Jan C. Recker, Antonie van Rensburg, Catherine Usoff, and Richard J. Welke. Business Process Management Education in Academia: Status, Challenges, and Recommendations. *Communications of the Association for Information Systems*, 27(1):743–776, 2010.

[BCHB12] Elizabeth A. Boyle, Thomas M. Connolly, Thomas Hainey, and James M. Boyle. Engagement in Digital Entertainment Games: A Systematic Review. *Computers in Human Behavior*, 28(3):771–780, 2012.

[BDH+09] Jörg Becker, Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In *Proceedings of the 17th European Conference on Information Systems (ECIS 2009)*, pages 2635–2647, Verona, Italy, 2009.

[BDJ+11] Giorgio Bruno, Frank Dengler, Ben Jennings, Rania Khalaf, Selmin Nurcan, Michael Prilla, Marcello Sarini, Rainer Schmidt, and António Rito Silva. Key Challenges for Enabling Agile BPM with Social Software. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(4):297–326, 2011.

[Ber67] Jacques Bertin. *Sémiologie Graphique. Les Diagrammes, les Réseaux, les Cartes.* Gauthier-Villars, Paris, France, 1st edition, 1967.

[BFV12] Marco Brambilla, Piero Fraternali, and Carmen Vaca. BPMN and Design Patterns for Engineering Social BPM Solutions. In *Proceedings of the 4th Workshop on Business Process Management and Social Software (BPMS2 2011)*, pages 219–230, Clermont-Ferrand, France, 2012.

[BGR05] Wasana Bandara, Guy G. Gable, and Michael Rosemann. Factors and Measures of Business Process Modelling: Model Building through a Multiple Case Study. *European Journal of Information Systems*, 14(4):347–360, 2005.

[BH98] Erik Brynjolfsson and Lorin M. Hitt. Beyond the Productivity Paradox: Computers are the Catalyst for Bigger Changes. *Communications of the ACM*, 41(8):49–55, 1998.

[BH04] Staffan Bjork and Jussi Holopainen. *Patterns in Game Design.* Charles River Media, Newton Centre, MA, United States, 1st edition, 2004.

[BHV+14] Mathieu Bastian, Matthew Hayes, William Vaughan, Sam Shah, Peter Skomoroch, Hyungjin Kim, Sal Uryasev, and Christopher Lloyd. LinkedIn Skills: Large-Scale Topic Extraction and Inference Mathieu. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys 2014)*, pages 1–8, Foster City, CA, United States, 2014.

[BICS07a] Wasana Bandara, Marta Indulska, Sandy Chong, and Shazia Sadiq. Major Issues in Business Process Management: An Expert Perspective. In *Proceedings of the 15th European Conference on Information Systems (ECIS 2007)*, pages 1240–1251, St. Gallen, Switzerland, 2007.

[BICS07b] Wasana Bandara, Marta Indulska, Sandy Chong, and Shazia Sadiq. Major Issues in Business Process Management: An Expert Perspective. In *Proceedings of the 15th European Conference on Information Systems (ECIS 2007)*, pages 1240–1251, St. Gallen, Switzerland, 2007.

[BJP11] Ilia Bider, Paul Johannesson, and Erik Perjons. A Strategy for Merging Social Software with Business Process Support. In *Proceedings of the 3rd Workshop on Business Process Management and Social Software (BPMS2 2010)*, pages 372–383, Hoboken, NJ, United States, 2011.

[BKR03] Jörg Becker, Martin Kugeler, and Michael Rosemann. *Process Management: A Guide for the Design of Business Processes*. Springer, Berlin, Heidelberg, Germany, 2003.

[BL95] Roy F. Baumeister and Mark R. Leary. The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation. *Psychological Bulletin*, 117(3):497–529, 1995.

[BL13] Ivo Blohm and Jan Marco Leimeister. Gamification: Design of IT-Based Enhancing Services for Motivational Support and Behavioral Change. *Business & Information Systems Engineering*, 5(4):275–278, 2013.

[BML05] Steve Benford, Carsten Magerkurth, and Peter Ljungstrand. Bridging the Physical and Digital in Pervasive Gaming. *Communications of the ACM*, 48(3):54–57, 2005.

[BMMM98] William H. Brown, Raphael C. Malveau, Hays W. McCormick, and Thomas J. Mowbray. *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. John Wiley & Sons, New York City, NY, United States, 1st edition, 1998.

[BNP10] Jörg Becker, Björn Niehaves, and Ralf Plattfaut. Stakeholder Involvement in Business Process Management Agenda-Setting and Implementation. In *Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010)*, Lima, Peru, 2010.

[Bog14] Ian Bogost. Why Gamification is Bullshit. In Steffen P Walz and Sebastian Deterding, editors, *The Gameful World*, chapter 2, pages 65–80. The MIT Press, Cambridge, MA, United States, 2014.

[Böh11] Martin Böhringer. Emergent Case Management for Ad-Hoc Processes: A Solution Based on Microblogging and Activity Streams. In *Proceedings of the 3rd Workshop on Business Process Management and Social Software (BPMS2 2010)*, pages 384–395, Hoboken, NJ, United States, 2011.

[Bos09] Barbaros Bostan. Player Motivations: A Psychological Perspective. *Computers in Entertainment*, 7(2):1–26, 2009.

[Bou12] Maura Bouqa. Mobile Communication, Gamification and Ludification. In *Proceedings of the 16th International Academic MindTrek Conference (MindTrek 2012)*, pages 295–301, Tampere, Finland, 2012.

[BP13] Jan Broer and Jens Poeppelbuss. Gamification – A New Phenomenon in Information Systems Research? In *Proceedings of the 24th Australasian Conference on Information Systems (ACIS 2013)*, Melbourne, Australia, 2013.

[BPV12] Jörg Becker, Wolfgang Probandt, and Oliver Vering. *Grundsätze Ordnungsmäßiger Modellierung: Konzeption und Praxisbeispiel für ein Effizientes Prozessmanagement.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2012.

[BR84] Michael L. Brodie and Dzenan Ridjanovic. On the Design and Specification of Database Transactions. In Michael L. Brodie, John Mylopoulos, and Joachim W. Schmidt, editors, *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, chapter 10, pages 277–312. Springer, Berlin, Heidelberg, Germany, 1984.

[BR12] Eike Bernhard and Jan C. Recker. Preliminary Insights from a Multiple Case Study on Process Modelling Impact. In *Proceedings of the 23rd Australasian Conference on Information Systems (ACIS 2012)*, Geelong, Australia, 2012.

[Bra13] Marco Brambilla. Application and Simplification of BPM Techniques for Personal Process Management. In *Proceedings of the 5th International Workshop on Business Process Management and Social Software (BPMS2 2012)*, pages 227–233, Tallinn, Estonia, 2013.

[Bri70] Richard W. Brislin. Back-Translation for Cross-Cultural Research. *Journal of Cross-Cultural Psychology*, 1(3):185–216, 1970.

[Bri86] Richard W. Brislin. The Wording and Translation of Research Instruments. In Walter J. Lonner and John W. Berry, editors, *Field Methods in Cross-Cultural Research*, chapter 5, pages 137–164. SAGE Publications, Thousand Oaks, CA, United States, 1st edition, 1986.

[Bro10] Ross A. Brown. Conceptual Modelling in 3D Virtual Worlds for Process Communication. In *Proceedings of the 7th Asia-Pacific Conference on Conceptual Modeling (APCCM 2010)*, pages 25–32, Brisbane, Australia, 2010.

[BRS95] Jörg Becker, Michael Rosemann, and Reinhard Schütte. Grundsätze ordnungsgemäßer Modellierung. *Wirtschaftsinformatik*, 37(5):435–445, 1995.

[BRSG07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The Aesthetics of Graph Visualization. In *Proceedings of the 3rd Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging (COMPAESTH 2007)*, pages 57–64, Banff, Canada, 2007.

[BRVB15] Max Bancora, Davide Ripamonti, Andrea Vaccarella, and Marco Brambilla. Model-Driven Development and Business Process Modeling Applied to Personal Productivity in the Consumer Mobile App Market. In *Proceedings of the 2nd ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft 2015)*, pages 174–175, Florence, Italy, 2015.

[BRvU00] Jörg Becker, Michael Rosemann, and Christoph von Uthmann. Guidelines of Business Process Modeling. In Wil M. P. van der Aalst, Jörg Desel, and Andreas Oberweis, editors, *Business Process Management: Models, Techniques, and Empirical Studies*, pages 30–49. Springer, Berlin, Heidelberg, Germany, 1st edition, 2000.

[BRW11] Ross A. Brown, Jan C. Recker, and Stephen West. Using Virtual Worlds for Collaborative Business Process Modeling. *Business Process Management Journal*, 17(3):546–564, 2011.

[BSP+94] Eric A. Bier, Maureen C. Stone, Ken Pier, Ken Fishkin, Thomas Baudelt, Matt Conway, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The See-Through Interface. In *Proceedings of the 1994 Conference on Human Factors in Computing Systems (CHI 1994)*, pages 445–446, Boston, MA, United States, 1994.

[Bun16] Bundesverband Interaktive Unterhaltungssoftware. Jahresreport der Computer- und Videospielbranche in Deutschland 2016. Technical report, Bundesverband Interaktive Unterhaltungssoftware, 2016.

[Bur14] Brian Burke. *Gamify: How Gamification Motivates People to Do Extraordinary Things.* Bibliomotion, Brookline, MA, United States, 1st edition, 2014.

[BVW15] An Bui, Daniel Veit, and Jane Webster. Gamification - A Novel Phenomenon or a New Wrapping for Existing Concepts? In *Proceedings of the 36th International Conference on Information Systems (ICIS 2015)*, pages 1–21, Fort Worth, TX, United States, 2015.

[Cai61] Roger Caillois. *Man, Play and Games.* University of Illinois Press, Champaign, IL, United States, 1st edition, 1961.

[CAN05] Mihály Csíkszentmihályi, Sami Abuhamdeh, and Jeanne Nakamura. Flow. In Andrew J. Elliot and Carol S. Dweck, editors, *Handbook of Competence and Motivation*, chapter 32, pages 598–608. The Guilford Press, New York City, NY, United States, 1st edition, 2005.

[Car05] Jorge Cardoso. Control-Flow Complexity Measurement of Processes and Weyuker's Properties. In *Proceedings of the 6th International Enformatika Conference (IEC 2005)*, pages 213–218, Budapest, Hungary, 2005.

[Car06] Jorge Cardoso. Process Control-Flow Complexity Metric: An Empirical Validation. In *Proceedings of the 3rd IEEE International Conference on Services Computing (SCC 2006)*, pages 167–173, Chicago, IL, United States, 2006.

[CBM+12] Thomas M. Connolly, Elizabeth A. Boyle, Ewan MacArthur, Thomas Hainey, and James M. Boyle. A Systematic Literature Review of Empirical Evidence on Computer Games and Serious Games. *Computers & Education*, 59(2):661–686, 2012.

[CCF13] Christopher Cheong, France Cheong, and Justin Filippou. Quick Quiz: A Gamified Approach for Enhancing Learning. In *Proceedings of the 17th Pacific Asia Conference on Information Systems (PACIS 2013)*, Jeju Island, South Korea, 2013.

[CCH+12] Joseph A. Cafazzo, Mark Casselman, Nathaniel Hamming, Debra K. Katzman, and Mark R. Palmert. Design of an mHealth App for the Self-Management of Adolescent Type 1 Diabetes: A Pilot Study. *Journal of Medical Internet Research*, 14(3):1–13, 2012.

[CCL+13] Timm Caporale, Murat Citak, Jonas Lehner, Andreas Schoknecht, and Meike Ullrich. Social BPM Lab: Characterization of a Collaborative Approach for Business Process Management Education. In *Proceedings of the 15th IEEE Conference on Business Informatics (CBI 2013)*, pages 367–373, Vienna, Austria, 2013.

[CF14a] Katheryn R. Christy and Jesse Fox. Leaderboards in a Virtual Classroom: A Test of Stereotype Threat and Social Comparison Explanations for Women's Math Performance. *Computers & Education*, 78:66–77, 2014.

[CF14b] Gregory W. Corder and Dale I. Foreman. *Nonparametric Statistics: A Step-by-Step Approach.* Wiley, Hoboken, NJ, United States, 2nd edition, 2014.

[CG09] Rachel Croson and Uri Gneezy. Gender Differences in Preferences. *Journal of Economic Literature*, 47(2):1–27, 2009.

[CGCC10] Eduardo H. Calvillo-Gámez, Paul Cairns, and Anna L. Cox. Assessing the Core Elements of the Gaming Experience. pages 3–7, 2010.

[CH93] Søren Christensen and Niels Damgaard Hansen. Coloured Petri Nets Extended With Place Capacities, Test Arcs and Inhibitor Arcs. In *Proceedings of the 14th International Conference on the Application and Theory of Petri Nets (Petri Nets 1993)*, pages 186–205, Chicago, IL, United States, 1993.

[CH04] WenShin Chen and Rudy Hirschheim. A Paradigmatic and Methodological Examination of Information Systems Research From 1991 to 2001. *Information Systems Journal*, 14(3):197–235, 2004.

[Che07] Jenova Chen. Flow in Games (and Everything Else). *Communications of the ACM*, 50(4):31–34, 2007.

[Cho15] Yu-Kai Chou. *Actionable Gamification: Beyond Points, Badges, and Leaderboards.* Octalysis Media, 2015.

[CKT+10] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. Predicting Protein Structures with a Multiplayer Online Ggame. *Nature*, 466(7307):756–760, 2010.

[CM55] Lee J. Cronbach and Paul E. Meehl. Construct Validity in Psychological Tests. *Psychological Bulletin*, 52(4):281–302, 1955.

[CM15] Christian Crumlish and Erin Malone. *Designing Social Interfaces: Principles, Patterns, and Practices for Improving The User Experience.* O'Reilly Media, Sebastopol, CA, United States, 2nd edition, 2015.

[CMKH12] Deborah R. Compeau, Barbara Marcolin, Helen Kelley, and Chris Higgins. Generalizability of Information Systems Research Using Student Subjects—A Reflection on Our Practices and Recommendations for Future Research. *Information Systems Research*, 23(4):1093–1109, 2012.

[CMNR06] Jorge Cardoso, Jan Mendling, Gustaf Neumann, and Hajo A. Reijers. A Discourse on Complexity of Process Models. In *Proceedings of the 2nd Workshop on Business Process Intelligence (BPI 2006)*, pages 117–128, Vienna, Austria, 2006.

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think.* Academic Press, San Diego, CA, United States, 1st edition, 1999.

[Coh92] Jacob Cohen. A Power Primer. *Psychological Bulletin*, 112(1):155–159, 1992.

[CP94] Judy Cameron and W. David Pierce. Reinforcement, Reward, and Intrinsic Motivation: A Meta-Analysis. *Review of Educational Research*, 64(3):363–423, 1994.

[Cre09] John W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* Sage Publications, Thousand Oaks, CA, United States, 3rd edition, 2009.

[Cro51] Lee J. Cronbach. Coefficient Alpha and the Internal Structure of Tests. *Psychometrika*, 16(3):297–334, 1951.

[Cro79] Philip B. Crosby. *Quality Is Free: The Art of Making Quality Certain.* McGraw-Hill, New York City, NY, United States, 1st edition, 1979.

[CS96] Michael K. Coleman and D. Stott Parker. Aesthetics-Based Graph Layout for Human Consumption. *Software: Practice and Experience*, 26(12):1415–1438, 1996.

[Csí90] Mihály Csíkszentmihályi. *Flow: The Psychology of Optimal Experience*. HarperCollins, New York City, NY, United States, 1st edition, 1990.

[CT12] Michele Chinosi and Alberto Trombetta. BPMN: An Introduction to the Standard. *Computer Standards and Interfaces*, 34(1):124–134, 2012.

[CVR+12] Jan Claes, Irene Vanderfeesten, Hajo A. Reijers, Jakob Pinggera, Matthias Weidlich, Stefan Zugal, Dirk Fahland, Barbara Weber, Jan Mendling, and Geert Poels. Tying Process Model Quality to the Modeling Process: The Impact of Structuring, Movement, and Speed. In *Proceedings of the 10th International Conference on Business Process Management (BPM 2012)*, pages 33–48, Tallinn, Estonia, 2012.

[DAJ11] Damien Djaouti, Julian Alvarez, and Jean-Pierre Jessel. Classifying Serious Games: The G/P/S Model. In Patrick Felicia, editor, *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*, chapter 6, pages 118–136. IGI Global, Hershey, PA, United States, 1st edition, 2011.

[Dau03] Berthold Daum. *Modeling Business Objects with XML Schema*. Morgan Kaufmann, San Diego, CA, United States, 2003.

[Dav89] Fred D. Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3):319–340, 1989.

[Dav93] Thomas H. Davenport. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, Boston, MA, United States, 1st edition, 1993.

[DB92] Paul Dourish and Victoria Bellotti. Awareness and Coordination in Shared Workspaces. In *Proceedings of the 4th ACM Conference on Computer Supported Cooperative Work (CSCW 1992)*, pages 107–114, Toronto, Canada, 1992.

[DBW89] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, 35(8):982–1003, 1989.

[DC11] Manoj Das and Linus Chow. Next Generation BPM Suites: Social and Collaborative. In Layna Fischer, editor, *Social BPM - Work, Planning and Collaboration Under the Impact of Social Technology*, chapter 18, pages 193–208. Future Strategies Inc., Lighthouse Point, FL, United States, 1st edition, 2011.

[DCT04] Heather Desurvire, Martin Caplan, and Jozsef A Toth. Using Heuristics to Evaluate the Playability of Games. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems (CHI 2004)*, pages 1509–1512, Vienna, Austria, 2004.

[DDKN11] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart E. Nacke. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference (MindTrek 2011)*, pages 9–15, Tampere, Finland, 2011.

[DDLM10] Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Henk Meijer. Area, Curve Complexity, and Crossing Resolution of Non-Planar Graph Drawings. In *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*, pages 15–20, Chicago, IL, United States, 2010.

[DDO08] Remco M. Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and Analysis of Business Process Models in BPMN. *Information and Software Technology*, 50(12):1281–1294, 2008.

[DDSV16] Johannes De Smedt, Jochen De Weerdt, Estefanía Serral, and Jan Vanthienen. Gamification of Declarative Process Models for Learning and Model Verification. In *Proceedings of the 3rd International Workshop on Decision Mining & Modeling for Business Processes (DeMiMoP 2015)*, pages 432–443, Innsbruck, Austria, 2016.

[DDvD+11] Remco M. Dijkman, Marlon Dumas, Boudewijn F. van Dongen, Reina Käärik, and Jan Mendling. Similarity of Business Process Models: Metrics and Evaluation. *Information Systems*, 36(2):498–516, 2011.

[DEL09] Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing Graphs with Right Angle Crossings. In *Proceedings of the 11th International Workshop on Algorithms and Data Structures (WADS 2009)*, pages 206–217, Banff, Canada, 2009.

[DeM82] Tom DeMarco. *Controlling Software Projects, Management Measurement & Estimation.* Yourdon Press, New York City, NY, United States, 1st edition, 1982.

[Dem00] William Edwards Deming. *Out of the Crisis.* The MIT Press, Cambridge, MA, United States, 2nd edition, 2000.

[Det12] Sebastian Deterding. Gamification: Designing for Motivation. *Interactions*, 19(4):14–17, 2012.

[Det13] Sebastian Deterding. Skill Atoms as Design Lenses for User-Centered Gameful Design. In *Proceedings of the CHI 2013 Workshop on Designing Gamification: Creating Gameful and Playful Experiences*, Paris, France, 2013.

[Det14] Sebastian Deterding. Eudamonic Design, Or: Six Invitations to Rethink Gamification. In Mathias Fuchs, Sonia Fizek, Paolo Ruffino, and Niklas Schrape, editors, *Rethinking Gamification*, chapter 15, pages 305–331. meson press, Lüneburg, Germany, 2014.

[Det15] Sebastian Deterding. The Lens of Intrinsic Skill Atoms: A Method for Gameful Design. *Human-Computer Interaction*, 30(3-4):294–335, 2015.

[DETT94] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for Drawing Graphs: An Annotated Bibliography. *Computational Geometry*, 4(5):235–282, 1994.

[DETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, Upper Saddle River, NJ, United States, 1st edition, 1999.

[DGK03] Jason Dedrick, Vijay Gurbaxani, and Kenneth L. Kraemer. Information Technology and Economic Performance: A Critical Review of the Empirical Evidence. *ACM Computing Surveys*, 35(1):1–28, 2003.

[DHLS09] Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Supporting Distributed Conceptual Modelling through Naming Conventions. *Enterprise Modelling and Information Systems Architectures*, 4(2):3–19, 2009.

[Die10] Reinhard Diestel. The Basics. In *Graph Theory*, chapter 1, pages 1–34. Springer, Berlin, Heidelberg, Germany, 4th edition, 2010.

[Dix11] Dan Dixon. Player Types and Gamification. In *Proceedings of the CHI 2011 Workshop on Gamification: Using Game Design Elements in NonGame Contexts*, pages 12–15, Vancouver, Canada, 2011.

[DKR99] Edward L. Deci, Richard Koestner, and Richard M. Ryan. A Meta-Analytic Review of Experiments Examining the Effects of Extrinsic Rewards on Intrinsic Motivation. *Psychological Bulletin*, 125(6):627–668, 1999.

[DLMR13] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2013.

[DM02] Berthold Daum and Udo Merten. *System Architecture with XML.* Morgan Kaufmann, San Francisco, CA, United States, 2002.

[DR85] Edward L. Deci and Richard M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior.* Springer Science+Business Media, New York City, NY, United States, 1st edition, 1985.

[DR87] Edward L. Deci and Richard M. Ryan. The Support of Autonomy and the Control of Behavior. *Journal of Personality and Social Psychology*, 53(6):1024–1037, 1987.

[DS09] Cody Dunne and Ben Shneiderman. Improving Graph Drawing Readability by Incorporating Readability Metrics: A Software Tool for Network Analysts. In *HCIL Tech Reports (HCIL-2009-13)*, pages 1–9, University of Maryland, College Park, MD, United States, 2009.

[DSDB15] Patrick Delfmann, Matthias Steinhorst, Hanns-Alexander Dietrich, and Jörg Becker. The Generic Model Query Language GMQL - Conceptual Specification, Implementation, and Runtime Evaluation. *Information Systems*, 47:129–177, 2015.

[DV11] Frank Dengler and Denny Vrandečić. Wiki-Based Maturing of Process Descriptions. In *Proceedings of the 9th International Conference on Business Process Management (BPM 2011)*, pages 313–328, Clermont-Ferrand, France, 2011.

[DW09] Heather Desurvire and Charlotte Wiberg. Game Usability Heuristics (PLAY) for Evaluating and Designing Better Games: The Next Iteration. In *Proceedings of the 3rd International Conference on Online Communities and Social Computing (OCSC 2009)*, pages 557–566, San Diego, CA, United States, 2009.

[Dzg16] Svetlana Dzgoeva. *Development of a Gamification Design Method for a Business Process Modeling Tool.* Master thesis, University of Münster, 2016.

[EB16] Frank Ebbers and Philipp Brune. The Authentication Game - Secure User Authentication by Gamification? In *Proceedings of the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016)*, pages 101–115, Ljubljana, Slovenia, 2016.

[EGH+10] Selim Erol, Michael Granitzer, Simone Happ, Sami Jantunen, Ben Jennings, Paul Johannesson, Agnes Koschmider, Selmin Nurcan, Davide Rossi, and Rainer Schmidt. Combining BPM and Social Software: Contradiction or Chance? *Journal of Software Maintenance and Evolution: Research and Practice*, 22(6-7):449–476, 2010.

[EJS10] Philip Effinger, Nicole Jogsch, and Sandra Seiz. On a Study of Layout Aesthetics for Business Process Models Using BPMN. In *Proceedings of the 2nd International Workshop on Business Process Modeling Notation (BPMN 2010)*, pages 31–45, Potsdam, Germany, 2010.

[ENST08] Simon Egenfeldt-Nielsen, Jonas Heide Smith, and Susana Pajares Tosca. *Understanding Video Games: The Essential Introduction.* Routledge, New York City, NY, United States, 1st edition, 2008.

[Ent16] Entertainment Software Association. Essential Facts About the Computer and Video Game Industry. Technical report, Entertainment Software Association, 2016.

[ES09] Holger Eichelberger and Klaus Schmid. Guidelines on the Aesthetic Quality of UML Class Diagrams. *Information and Software Technology*, 51(12):1686–1698, 2009.

[FD11] Kathrin Figl and Michael Derntl. The Impact of Perceived Cognitive Effectiveness on Perceived Usefulness of Visual Conceptual Modeling Languages. In *Proceedings of the 30th International Conference on Conceptual Modeling (ER 2011)*, pages 78–91, Brussels, Belgium, 2011.

[FDM+08] Rosta Farzan, Joan M. DiMicco, David R. Millen, Beth Brownholtz, Werner Geyer, and Casey Dugan. Results from Deploying a Participation Incentive Mechanism within the Enterpris. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI 2008)*, pages 563–572, Florence, Italy, 2008.

[Fed02] Melissa A. Federoff. *Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games.* Master thesis, Indiana University, 2002.

[Fer07a] Christopher J. Ferguson. Evidence for Publication Bias in Video Game Violence Effects Literature: A Meta-Analytic Review. *Aggression and Violent Behavior*, 12(4):470–482, 2007.

[Fer07b] Christopher J. Ferguson. The Good, the Bad and the Ugly: A Meta-Analytic Review of Positive and Negative Effects of Violent Video Games. *Psychiatric Quarterly*, 78(4):309–316, 2007.

[FFRS14] Mathias Fuchs, Sonia Fizek, Paolo Ruffino, and Nikl Schrape. *Rethinking Gamification.* meson press, Lüneburg, Germany, 1st edition, 2014.

[FHH+93] Michael Formann, Torben Hagerup, James Haralambides, Michael Kaufmann, Frank T. Leighton, Antonios Simvonis, Emo Welzl, and Gerhard J. Woeginger. Drawing Graphs in the Plane with High Resolution. *SIAM Journal on Computing*, 22(5):1035–1052, 1993.

[Fie09] Andy Field. *Discovering Statistics Using SPSS.* Sage, London, United Kingdom, 3rd edition, 2009.

[Fig12] Kathrin Figl. Symbol Choice and Memory of Visual Models. In *Proceedings of the 28th IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012)*, pages 97–100, Innsbruck, Austria, 2012.

[FK10] Christopher J. Ferguson and John Kilburn. Much Ado About Nothing: The Misestimation and Overinterpretation of Violent Video Game Effects in Eastern and Western Nations: Comment on Anderson et al. (2010). *Psychological Bulletin*, 136(2):174–178, 2010.

[FL11] Kathrin Figl and Ralf Laue. Cognitive Complexity in Business Process Modeling. In *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*, pages 452–466, London, United Kingdom, 2011.

[FL15] Kathrin Figl and Ralf Laue. Influence Factors for Local Comprehensibility of Process Models. *International Journal of Human-Computer Studies*, 82:96–110, 2015.

[Fle48] Rudolf Flesch. A New Readability Yardstick. *Journal of Applied Psychology*, 32(3):221–233, 1948.

[FMP11] Fabian Friedrich, Jan Mendling, and Frank Puhlmann. Process Model Generation from Natural Language Text. In *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*, pages 482–496, London, United Kingdom, 2011.

[FMR12] Catherine O. Fritz, Peter E. Morris, and Jennifer J. Richler. Effect Size Estimates: Current Use, Calculations, and Interpretation. *Journal of Experimental Psychology: General*, 141(1):2–18, 2012.

[FMS09] Kathrin Figl, Jan Mendling, and Mark Strembeck. Towards a Usability Assessment of Process Modeling Languages. In *Proceedings of the 8th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2009)*, pages 118–136, Berlin, Germany, 2009.

[FMS13] Kathrin Figl, Jan Mendling, and Mark Strembeck. The Influence of Notational Deficiencies on Process Model Comprehension. *Journal of the Association for Information Systems*, 14(6):312–338, 2013.

[FMSR10] Kathrin Figl, Jan Mendling, Mark Strembeck, and Jan C. Recker. On the Cognitive Effectiveness of Routing Symbols in Process Modeling Languages. In *Proceedings of the 13th International Conference on Business Information Systems (BIS 2010)*, pages 230–241, Berlin, Germany, 2010.

[For10] Santo Fortunato. Community Detection in Graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[Fow04] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Pearson Education, Boston, MA, United States, 3rd edition, 2004.

[Fox05] Brent Fox. *Game Interface Design*. Thomson Course Technology, Boston, MA, United States, 1st edition, 2005.

[FR95] Christina M. Frederick and Richard M. Ryan. Self-Determination in Sport: A Review Using Cognitive Evaluation Theory. *International Journal of Sport Psychology*, 26(1):5–23, 1995.

[FR16a] Kathrin Figl and Jan C. Recker. Exploring Cognitive Style and Task-Specific Preferences for Process Representations. *Requirements Engineering*, 21(1):63–85, 2016.

[FRM13] Kathrin Figl, Jan C. Recker, and Jan Mendling. A Study on the Effects of Routing Symbol Design on Process Model Comprehension. *Decision Support Systems*, 54(2):1104–1118, 2013.

[Fry68] Edward Fry. A Readability Formula That Saves Time. *Journal of Reading*, 11(7):513–516, 575–578, 1968.

[FS14] Kathrin Figl and Mark Strembeck. On the Importance of Flow Direction in Business Process Models. In *Proceedings of the 9th International Conference on Software Engineering and Applications (ICSOFT-EA 2014)*, pages 132–136, Vienna, Austria, 2014.

[FS15] Kathrin Figl and Mark Strembeck. Findings from an Experiment on Flow Direction of Business Process Models. In *Proceedings of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2015)*, pages 59–73, Innsbruck, Austria, 2015.

[Fuc14] Mathias Fuchs. Predigital Precursors of Gamification. In Mathias Fuchs, Sonia Fizek, Paolo Ruffino, and Niklas Schrape, editors, *Rethinking Gamification*, chapter 6, pages 119–140. meson press, Lüneburg, Germany, 2014.

[Ful08] Tracy Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games.* Morgan Kaufmann Publishers, Burlington, MA, United States, 1st edition, 2008.

[FvdW06] P. J. M. Frederiks and Theo P. van der Weide. Information Modeling: The Process and the Required Competencies of its Participants. *Data & Knowledge Engineering*, 58(1):4–20, 2006.

[FWG14] Lauren S. Ferro, Steffen P. Walz, and Stefan Greuter. Gamicards - An Alternative Method for Paper-Prototyping the Design of Gamified Systems. In *Proceedings of the 13th International Conference on Entertainment Computing (ICEC 2014)*, pages 11–18, Sydney, Australia, 2014.

[GB04] Erich Gamma and Kent Beck. *Eclipse erweitern: Prinzipien, Patterns und Plug-Ins.* Addison-Wesley, Munich, Germany, 1st edition, 2004.

[GBR12] Hanwen Guo, Ross A. Brown, and Rune K. Rasmussen. Virtual Worlds as a Model-View Approach to the Communication of Business Processes Models.

In *Proceedings of the CAiSE 2012 Forum at the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012)*, pages 66–73, Gdańsk, Poland, 2012.

[GD05] Marylène Gagné and Edward L. Deci. Self-Determination Theory and Work Motivation. *Journal of Organizational Behavior*, 26(4):331–362, 2005.

[Gee07] James Paul Gee. *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, London, United Kingdom, 2nd edition, 2007.

[GGR96] Carl Gutwin, Saul Greenberg, and Mark Roseman. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In Martina Angela Sasse, R. Jim Cunningham, and Russel L. Winder, editors, *People and Computers X: Proceedings of the 1996 International BCS Human Computer Interaction Conference (HCI 1996)*, chapter 18, pages 281–298. Springer, Berlin, Heidelberg, Germany, 1996.

[GHA10] Nicolas Genon, Patrick Heymans, and Daniel Amyot. Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In *Proceedings of the 3rd International Conference on Software Language Engineering (SLE 2010)*, pages 377–396, Eindhoven, Netherlands, 2010.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, MA, United States, 1995.

[GHS95] Dimitrios Georgakopoulos, Mark Hornick, and Amit Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[Gia01] George M. Giaglis. A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques. *International Journal of Flexible Manufacturing Systems*, 13(2):209–228, 2001.

[Gil10] Phil Gilbert. The Next Decade of BPM. In *Proceedings of the 8th International Conference on Business Process Management (BPM 2010)*, pages 1–4, Hoboken, NJ, United States, 2010.

[GJ83] Michael R. Garey and David S. Johnson. Crossing Number is NP-Complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.

[GL06] Volker Gruhn and Ralf Laue. Complexity Metrics for Business Process Models. In *Proceedings of the 9th International Conference on Business Information Systems (BIS 2006)*, pages 1–12, Klagenfurt, Austria, 2006.

[GL07] Volker Gruhn and Ralf Laue. What Business Process Modelers Can Learn from Programmers. *Science of Computer Programming*, 65(1):4–13, 2007.

[Gla68] P. G. W. Glare. *Oxford Latin Dictionary*. Oxford University Press, Oxford, United Kingdom, 1st edition, 1968.

[GM99] Carsten Gutwenger and Petra Mutzel. Planar Polyline Drawings with Good Angular Resolution. In *Proceedings of the 6th International Symposium on Graph Drawing (GD 1998)*, pages 167–182, Montreal, Canada, 1999.

[Goe13] Geoff Goehle. Gamification and Web-based Homework. *PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Students*, 23(3):234–246, 2013.

[Goo10] C. James Goodwin. *Research in Psychology: Methods and Design*. Wiley, Hoboken, NJ, United States, 6th edition, 2010.

[GPR95] Narasimhaiah Gorla, Hao Che Pu, and Walter O. Rom. Evaluation of Process Tools in Systems Analysis. *Information and Software Technology*, 37(2):119–126, 1995.

[GRS12] Chiara Ghidini, Marco Rospocher, and Luciano Serafini. Conceptual Modeling in Wikis: A Reference Architecture and a Tool. In *Proceedings of the 4th International Conference on Information, Process, and Knowledge Management (eKNOW 2012)*, pages 128–135, Valencia, Spain, 2012.

[GS14] Samuel B. Green and Neil J. Salkind. *Using SPSS for Windows and Macintosh: Analyzing and Understanding Data*. Pearson, Upper Saddle River, NJ, United States, 7th edition, 2014.

[GS16] Gilles E. Gignac and Eva T. Szodorai. Effect Size Guidelines for Individual Differences Researchers. *Personality and Individual Differences*, 102:74–78, 2016.

[GSB00] David Gefen, Detmar Straub, and Marie-Claude Boudreau. Structural Equation Modeling and Regression: Guidelines for Research Practice. *Communications of the AIS*, 4, 2000.

[GSSM16] Elena Gorbacheva, Armin Stein, Theresa Schmiedel, and Oliver Müller. The Role of Gender in Business Process Management Competence Supply. *Business & Information Systems Engineering*, 58(3):213–231, 2016.

[GT94] Ashim Garg and Roberto Tamassia. Planar Drawings and Angular Resolution: Algorithms and Bounds. In *Proceedings of the 2nd Annual European Symposium on Algorithms (ESA 1994)*, pages 12–23, Utrecht, Netherlands, 1994.

[GT98] Dimitrios Georgakopoulos and Aphrodite Tsalgatidou. Technology and Tools for Comprehensive Business Process Lifecycle Management. In Asuman

Doğaç, Leonid Kalinichenko, M. Tamer Özsu, and Amit Sheth, editors, *Workflow Management Systems and Interoperability*, chapter 16, pages 356–395. Springer, Berlin, Heidelberg, Germany, 1998.

[GV03] Claude Girault and Rüdiger Valk. *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2003.

[Ham10] Michael Hammer. What is Business Process Management? In Jan vom Brocke and Michael Rosemann, editors, *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, chapter 1, pages 3–16. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[Ham17] Juho Hamari. Do Badges Increase User Activity? A Field Experiment on the Effects of Gamification. *Computers in Human Behavior*, 71:469–478, 2017.

[HC93] Michael Hammer and James Champy. *Reengineering the Corporation: A Manifesto for Business Revolution.* Harper Business Books, New York City, NY, United States, 1st edition, 1993.

[HE11] Juho Hamari and Veikko Eranti. Framework for Designing and Evaluating Game Achievements. In *Proceedings of the 5th International Conference of the Digital Games Research Association: Think, Design, Play (DiGRA 2011)*, Utrecht, Netherlands, 2011.

[Her14] Mario Herger. *Enterprise Gamification: Engaging People by Letting Them Have Fun.* CreateSpace Independent Publishing, North Charleston, SC, United States, 1st edition, 2014.

[Hev07] Alan R. Hevner. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2):87–92, 2007.

[HF15] Michael D. Hanus and Jesse Fox. Assessing the Effects of Gamification in the Classroom: A Longitudinal Study on Intrinsic Motivation, Social Comparison, Satisfaction, Effort, and Academic Performance. *Computers & Education*, 80:152–161, 2015.

[HFL+11] Constantin Houy, Peter Fettke, Peter Loos, Wil M. P. van der Aalst, and John Krogstie. Business Process Management in the Large. *Business & Information Systems Engineering*, 3(6):385–388, 2011.

[HFL12] Constantin Houy, Peter Fettke, and Peter Loos. Understanding Understandability of Conceptual Models - What Are We Actually Talking About? In *Proceedings of the 31st International Conference on Conceptual Modeling (ER 2012)*, pages 64–77, Florence, Italy, 2012.

[HFL14] Constantin Houy, Peter Fettke, and Peter Loos. On the Theoretical Foundations of Research Into the Understandability of Business Process Models. In *Proceedings of the 22nd European Conference on Information Systems (ECIS 2014)*, Tel Aviv, Israel, 2014.

[HH12] Kai Huotari and Juho Hamari. Defining Gamification - A Service Marketing Perspective. In *Proceedings of the 16th International Academic MindTrek Conference (MindTrek 2012)*, pages 17–22, Tampere, Finland, 2012.

[HHE08] Weidong Huang, Seok-Hee Hong, and Peter Eades. Effects of Crossing Angles. In *Proceedings of the 2008 IEEE Pacific Visualization Symposium (PacificVIS 2008)*, pages 41–46, Kyoto, Japan, 2008.

[HIHK14] Lassi Haaranen, Petri Ihantola, Lasse Hakulinen, and Ari Korhonen. How (not) to Introduce Badges to Online Exercises. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE 2014)*, pages 33–38, Atlanta, GA, United States, 2014.

[Hip06] Hajo Hippner. Bedeutung, Anwendungen und Einsatzpotenziale von Social Software. *HMD - Praxis der Wirtschaftsinformatik*, 43(252):6–16, 2006.

[HJS91] Peter Huber, Kurt Jensen, and Robert M. Shapiro. Hierarchies in Coloured Petri Nets. In *Proceedings of the 10th International Conference on Applications and Theory of Petri Nets: Advances in Petri Nets 1990 (ICAPTN 1989)*, pages 313–341, Bonn, Germany, 1991.

[HK14] Juho Hamari and Jonna Koivisto. Measuring Flow in Gamification: Dispositional Flow Scale-2. *Computers in Human Behavior*, 40:133–143, 2014.

[HK15] Juho Hamari and Jonna Koivisto. "Working Out for Likes": An Empirical Study on Social Influence in Exercise Gamification. *Computers in Human Behavior*, 50:333–347, 2015.

[HK17] Juho Hamari and Lauri Keronen. Why do people play games? A meta-analysis. *International Journal of Information Management*, 37(3):125–141, 2017.

[HKS14] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does Gamification Work? — A Literature Review of Empirical Studies on Gamification. In *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS 2014)*, pages 3025–3034, Waikoloa, HI, United States, 2014.

[HL14] Farideh Heidari and Pericles Loucopoulos. Quality Evaluation Framework (QEF): Modeling and Evaluating Quality of Business Processes. *International Journal of Accounting Information Systems*, 15(3):193–223, 2014.

[HLZ04] Robin Hunicke, Marc LeBlanc, and Robert Zubek. MDA: A Formal Approach to Game Design and Game Research. In *Proceedings of the 2004 AAAI Workshop on Challenges in Game Artificial Intelligence*, San Jose, CA, United States, 2004.

[HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science Research in Information Systems. *MIS Quarterly*, 28(1):75–105, 2004.

[HPS13] Seyed Alireza Hajimirsadeghi, Hye-Young Paik, and John Shepherd. Processbook: Towards Social Network-Based Personal Process Management. In *Proceedings of the 5th Workshop on Business Process Management and Social Software (BPMS2 2012)*, pages 268–279, Tallinn, Estonia, 2013.

[HS06] Mikel Harry and Richard Schroeder. *Six Sigma: Tthe Breakthrough Management Strategy Revolutionizing the World's Top Corporations.* Crown Business, 2006.

[HS09] Stijn J. B. A. Hoppenbrouwers and Bart Schotten. A Game Prototype for Basic Process Model Elicitation. In *Proceedings of the 2nd IFIP Working Conference on The Practice of Enterprise Modeling (PoEM 2009)*, pages 222–236, Stockholm, Sweden, 2009.

[HSR+16] Juho Hamari, David J. Shernoff, Elizabeth Rowe, Brianno Coller, Jodi Asbell-Clarke, and Teon Edwards. Challenging Games Help Students Learn: An Empirical Study on Engagement, Flow and Immersion in Game-based Learning. *Computers in Human Behavior*, 54:170–179, 2016.

[HT85] Starr Roxanne Hiltz and Murray Turoff. Structuring Computer-Mediated Communication Systems to Avoid Information Overload. *Communications of the ACM*, 28(7):680–689, 1985.

[HT14] Juho Hamari and Janne Tuunanen. Player Types: A Meta-Synthesis. *Transactions of the Digital Games Research Association*, 1(2):29–53, 2014.

[Hua07] Weidong Huang. Using Eye Tracking to Investigate Graph Layout Effects. In *Proceedings of the 6th International Asia-Pacific Symposium on Visualization (APVIS 2007)*, pages 97–100, Sydney, Australia, 2007.

[Hui49] Johan Huizinga. *Homo Ludens: A Study of the Play-Element in Culture.* Routledge & Kegan Paul, London, United Kingdom, 2nd edition, 1949.

[Hwa05] Yujong Hwang. Investigating Enterprise Systems Adoption: Uncertainty Avoidance, Intrinsic Motivation, and the Technology Acceptance Model. *European Journal of Information Systems*, 14(2):150–161, 2005.

[IAT16] IATF. *IATF 16949:2016 Anforderungen an Qualitätsmanagementsysteme für die Serien- und Ersatzteilproduktion in der Automobilindustrie.* International Automotive Task Force, 1st edition, 2016.

[IC07] Yavuz Inal and Kursat Cagiltay. Flow Experiences of Children in an Interactive Social Game Environment. *British Journal of Educational Technology*, 38(3):455–464, 2007.

[IEH99] Il Im, Omar A. El Sawy, and Alexander Hars. Competence and Impact of Tools for BPR. *Information & Management*, 36(6):301–311, 1999.

[IGRR09] Marta Indulska, Peter Green, Jan C. Recker, and Michael Rosemann. Business Process Modeling: Perceived Benefits. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER 2009)*, pages 458–471, Gramado, Brazil, 2009.

[IRRG09] Marta Indulska, Jan C. Recker, Michael Rosemann, and Peter Green. Business Process Modeling: Current Issues and Future Challenges. In *Proceedings of the 21st International Conference on Advanced Information Systems Engineering (CAiSE 2009)*, pages 501–514, Amsterdam, Netherlands, 2009.

[ISO15a] ISO. *ISO 9000:2015 Quality management systems – Fundamentals and vocabulary.* International Organization for Standardization, 4th edition, 2015.

[ISO15b] ISO. *ISO 9001:2015 Quality management systems – Requirements.* International Organization for Standardization, 5th edition, 2015.

[JAW09] Paul Johannesson, Birger Andersson, and Petia Wohed. Business Process Management with Social Software Systems – A New Paradigm for Work Organisation. In *Proceedings of the 1st Workshop on Business Process Management and Social Software (BPMS2 2008)*, pages 659–665, Milano, Italy, 2009.

[JE02] Susan A. Jackson and Robert C. Eklund. Assessing Flow in Physical Activity: The Flow State Scale-2 and Dispositional Flow Scale-2. *Journal of Sport & Exercise Psychology*, 24(2):133–150, 2002.

[Jen91] Kurt Jensen. Coloured Petri Nets: A High Level Language for System Design and Analysis. In *Proceedings of the 10th International Conference on Applications and Theory of Petri Nets: Advances in Petri Nets 1990 (ICAPTN 1989)*, pages 342–416, Bonn, Germany, 1991.

[Jen97] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use (Volume 3).* Springer, Berlin, Heidelberg, Germany, 1st edition, 1997.

[JG99] Joseph M. Juran and A. Blanton Godfrey. *Juran's Quality Handbook.* McGraw-Hill, New York City, NY, United States, 5th edition, 1999.

[Joh13] Valen E Johnson. Revised Standards for Statistical Evidence. *Proceedings of the National Academy of Sciences of the United States of America*, 110(48):19313–19317, 2013.

[Jör69] Karl G. Jöreskog. A General Approach to Confirmatory Maximum Likelihood Factor Analysis. *Psychometrika*, 34(2):183–202, 1969.

[Jør12] Kristine Jørgensen. Between the Game System and the Fictional World: A Study of Computer Game Interfaces. *Games and Culture*, 7(2):142–163, 2012.

[Juu11] Jesper Juul. *Half-Real: Video Games between Real Rules and Fictional Worlds*. MIT Press, Cambridge, MA, United States, 2011.

[KAEM13] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, and William Money. Big Data: Issues and Challenges Moving Forward. In *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS 2013)*, pages 995–1004, Maui, HI, United States, 2013.

[KAL08] Jeffrey Y. Kim, Jonathan P. Allen, and Elan Lee. Alternate Reality Gaming. *Communications of the ACM*, 51(2):36–42, 2008.

[KBM14] Karl M. Kapp, Lucas Blair, and Rich Mesch. *The Gamification of Learning and Instruction Fieldbook: Ideas into Practice*. Wiley, Hoboken, NJ, United States, 2014.

[Kem11] Sandy Kemsley. Leveraging Social BPM for Enterprise Transformation. In Layna Fischer, editor, *Social BPM - Work, Planning and Collaboration Under the Impact of Social Technology*, chapter 7, pages 77–84. Future Strategies Inc., Lighthouse Point, FL, United States, 1st edition, 2011.

[KF99] John Kowal and Michelle S. Fortier. Motivational Determinants of Flow: Contributions From Self-Determination Theory. *The Journal of Social Psychology*, 139(3):355–368, 1999.

[KFS15] Agnes Koschmider, Kathrin Figl, and Andreas Schoknecht. A Comprehensive Overview of Visual Design of Process Model Element Labels. In *Proceedings of the 4th International Workshop on the Theory and Application of Visualizations and Human-Centric Aspects in Processes (TAProViz 2015)*, pages 571–582, Innsbruck, Austria, 2015.

[KH14] Jonna Koivisto and Juho Hamari. Demographic Differences in Perceived Benefits from Gamification. *Computers in Human Behavior*, 35:179–188, 2014.

[KJ03] John Krogstie and Håvard D. Jørgensen. Quality of Interactive Models. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*, pages 351–363, Tampere, Finland, 2003.

[KJL10] Won Kim, Ok-Ran Jeong, and Sang-Won Lee. On Social Web Sites. *Information Systems*, 35(2):215–236, 2010.

[KK02] Dimitris Karagiannis and Harald Kühn. Metamodelling Platforms. In *Proceedings of the 3rd International Conference on E-Commerce and Web Technologies (EC-Web 2002)*, page 182, Aix-en-Provence, France, 2002.

[Kli11] Rex B. Kline. *Principles and Practice of Structural Equation Modeling*. The Guilford Press, New York City, NY, United States, 3rd edition, 2011.

[KLL09] Ryan K. L. Ko, Stephen S.G. Lee, and Eng Wah Lee. Business Process Management (BPM) Standards: A Survey. *Business Process Management Journal*, 15(5):744–791, 2009.

[KLS95] John Krogstie, Odd I. Lindland, and Guttorm Sindre. Defining Quality Aspects for Conceptual Models. In Eckhard D. Falkenberg, Wolfgang Hesse, and Antoni Olivé, editors, *Information Systems Concepts: Towards a Consolidation of Views*, chapter 22, pages 216–231. Springer, Cham, Switzerland, 1st edition, 1995.

[KM96] Nereu F. Kock and Robert J. McQueen. Product Flow, Breadth and Complexity of Business Processes: An Empirical Study of 15 Business Processes in Three Organizations. *Business Process Re-engineering & Management Journal*, 2(2):8–22, 1996.

[Kol33] Andrey Kolmogorov. Sulla Determinazione Empirica di Una Legge di Distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91, 1933.

[Kos05] Raph Koster. *A Theory of Fun for Game Design*. Paraglyph Press, Scottsdale, AZ, United States, 1st edition, 2005.

[KP78] Brian W. Kernighan and P. J. Plauger. *The Elements of Programming Style*. McGraw-Hill, New York City, NY, United States, 2nd edition, 1978.

[KR09] Michael Koch and Alexander Richter. *Enterprise 2.0 - Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen*. Oldenbourg Wissenschaftsverlag, Munich, Germany, 2nd edition, 2009.

[KRM16] Tyge-F. Kummer, Jan C. Recker, and Jan Mendling. Enhancing Understandability of Process Models through Cultural-Dependent Color Adjustments. *Decision Support Systems*, 87:1–12, 2016.

[Kro12] John Krogstie. *Model-Based Development and Evolution of Information Systems: A Quality Approach*. Springer, Berlin, Heidelberg, Germany, 1st edition, 2012.

[Kro16] John Krogstie. *Quality in Business Process Modeling*. Springer, Berlin, Heidelberg, Germany, 1st edition, 2016.

[KSJ06] John Krogstie, Guttorm Sindre, and Håvard D. Jørgensen. Process Models Representing Knowledge for Action: A Revised Quality Framework. *European Journal of Information Systems*, 15(1):91–102, 2006.

[KSR09] Agnes Koschmider, Minseok Song, and Hajo A. Reijers. Social Software for Modeling Business Processes. In *Proceedings of the 1st Workshop on Business Process Management and Social Software (BPMS2 2008)*, pages 666–677, Milano, Italy, 2009.

[KTCK12] Atreyi Kankanhalli, Mahdieh Taher, Huseyin Cavusoglu, and Seung Hyun Kim. Gamification: A New Paradigm for Online User Engagement. In *Proceedings of the 33rd International Conference on Information Systems (ICIS 2012)*, Orlando, FL, United States, 2012.

[KTD+13] Pavel Kucherbaev, Stefano Tranquillini, Florian Daniel, Fabio Casati, Maurizio Marchese, Marco Brambilla, and Piero Fraternali. Business Processes for the Crowd Computer. In *Proceedings of the 5th Workshop on Business Process Management and Social Software (BPMS2 2012)*, pages 256–267, Tallinn, Estonia, 2013.

[KtHvdA03] Bartek Kiepuszewski, Arthur H. M. ter Hofstede, and Wil M. P. van der Aalst. Fundamentals of Control Flow in Workflow. *Acta Informatica*, 39(3):143–209, 2003.

[KUHO15] Agnes Koschmider, Meike Ullrich, Antje Heine, and Andreas Oberweis. Revising the Vocabulary of Business Process Element Labels. In *Proceedings of the 27th Conference on Advanced Information Systems Engineering (CAiSE 2015)*, pages 69–83, Stockholm, Sweden, 2015.

[LB13] Ralf Laue and Michael Becker. Evaluating Social Tagging for Business Process Models. In *Proceedings of the 5th Workshop on Business Process Management and Social Software (BPMS2 2012)*, pages 280–291, Tallinn, Estonia, 2013.

[LBC15] Richard N. Landers, Kristina N. Bauer, and Rachel C. Callan. Gamification of Task Performance with Leaderboards: A Goal Setting Experiment. *Computers in Human Behavior*, 71:508–515, 2015.

[LE06] Yair Levy and Timothy J. Ellis. A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. *Informing Science*, 9:181–212, 2006.

[Leh97] Erich L. Lehmann. *Testing Statistical Hypotheses.* Springer Science+Business Media, New York City, NY, United States, 2nd edition, 1997.

[Len03] Kirsten Lenz. *Modellierung und Ausführung von E-Business-Prozessen mit XML-Netzen.* Verlag für Wissenschaft und Forschung, Berlin, Germany, 1st edition, 2003.

[Leo13] Henrik Leopold. *Natural Language in Business Process Models: Theoretical Foundations, Techniques, and Applications.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2013.

[LESM$^+$13] Henrik Leopold, Rami-Habib Eid-Sabbagh, Jan Mendling, Leonardo Guerreiro Azevedo, and Fernanda Baiao. Detection of Naming Convention Violations in Process Models for Different Languages. *Decision Support Systems*, 56:310–325, 2013.

[Lev60] Howard Levene. Robust Tests for Equality of Variances. In Ingram Olkin, Sudish G. Ghurye, Wassily Hoeffding, William G. Madow, and Henry B. Mann, editors, *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 278–292. Stanford University Press, Stanford, CA, United States, 1st edition, 1960.

[LGN73] Mark R. Lepper, David Greene, and Richard E. Nisbett. Undermining Children's Intrinsic Interest with Extrinsic Reward: A Test of the Overjustification Hypothesis. *Journal of Personality and Social Psychology*, 28(1):129–137, 1973.

[LK06] Beate List and Birgit Korherr. An Evaluation of Conceptual Business Process Modelling Languages. In *Proceedings of the 21st ACM Symposium on Applied Computing (SAC 2006)*, pages 1532–1539, Dijon, France, 2006.

[LL02] Edwin A. Locke and Gary P. Latham. Building a Practically Useful Theory of Goal Setting and Task Motivation: A 35-Year Odyssey. *American Psychologist*, 57(9):705–717, 2002.

[LL06] Edwin A. Locke and Gary P. Latham. New Directions in Goal-Setting Theory. *Current Directions in Psychological Science*, 15(5):265–268, 2006.

[LMG16] Henrik Leopold, Jan Mendling, and Oliver Gunther. Learning from Quality Issues of BPMN Models from Industry. *IEEE Software*, 33(4):26–33, 2016.

[LMP12] Henrik Leopold, Jan Mendling, and Artem Polyvyanyy. Generating Natural Language Texts from Business Process Models. In *Proceedings of the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012)*, pages 64–79, Gdańsk, Poland, 2012.

[LMP14] Henrik Leopold, Jan Mendling, and Artem Polyvyanyy. Supporting Process Model Validation through Natural Language Generation. *IEEE Transactions on Software Engineering*, 40(8):818–840, 2014.

[LMR11] Henrik Leopold, Jan Mendling, and Hajo A. Reijers. On the Automatic Labeling of Process Models. In *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE 2011)*, pages 512–520, London, United Kingdom, 2011.

[LMRL14] Henrik Leopold, Jan Mendling, Hajo A. Reijers, and Marcello La Rosa. Simplifying Process Model Abstraction: Techniques for Generating Model Names. *Information Systems*, 39:134–151, 2014.

[LO01] Kirsten Lenz and Andreas Oberweis. Modeling Interorganizational Workflows with XML Nets. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS 2001)*, Maui, HI, United States, 2001.

[Loh15] Matthias J. Lohrmann. *Business Process Quality Management*. Doctoral thesis, Universität Ulm, 2015.

[LRJA10] Victor E. Lee, Ning Ruan, Ruoming Jin, and Charu C. Aggarwal. A Survey of Algorithms for Dense Subgraph Discovery. In Charu C. Aggarwal and Haixun Wang, editors, *Managing and Mining Graph Data*, chapter 10, pages 303–336. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[LRvdA+11] Marcello La Rosa, Hajo A. Reijers, Wil M. P. van der Aalst, Remco M. Dijkman, Jan Mendling, Marlon Dumas, and Luciano García-Bañuelos. APROMORE: An Advanced Process Model Repository. *Expert Systems with Applications*, 38(6):7029–7040, 2011.

[LS87] Jill H. Larkin and Herbert A. Simon. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11(1):65–99, 1987.

[LSM09] Henrik Leopold, Sergey Smirnov, and Jan Mendling. On Labeling Quality in Business Process Models. In *Proceedings of the 8th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2009)*, pages 42–57, Berlin, Germany, 2009.

[LSM10] Henrik Leopold, Sergey Smirnov, and Jan Mendling. Refactoring of Process Model Activity Labels. In *Proceedings of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB 2010)*, pages 268–276, Cardiff, United Kingdom, 2010.

[LSM11] Henrik Leopold, Sergey Smirnov, and Jan Mendling. Recognising Activity Labeling Styles in Business Process Models. *Enterprise Modelling and Information Systems Architectures*, 6(1):16–29, 2011.

[LSM12] Henrik Leopold, Sergey Smirnov, and Jan Mendling. On the Refactoring of Activity Labels in Business Process Models. *Information Systems*, 37(5):443–459, 2012.

[LSS94] Odd I. Lindland, Guttorm Sindre, and Arne Sølvberg. Understanding Quality in Conceptual Modeling. *IEEE Software*, 11(2):42–49, 1994.

[LSSL81] Edwin A. Locke, Karyll N. Shaw, Lise M. Saari, and Gary P. Latham. Goal Setting and Task Performance: 1969-1981. *Psychological Bulletin*, 90(1):125–152, 1981.

[LtHW+11] Marcello La Rosa, Arthur H. M. ter Hofstede, Petia Wohed, Hajo A. Reijers, Jan Mendling, and Wil M. P. van der Aalst. Managing Process Model Complexity Via Concrete Syntax Modifications. *IEEE Transactions on Industrial Informatics*, 7(2):255–265, 2011.

[LWC+14] Cameron Lister, Joshua H. West, Ben Cannon, Tyler Sax, and David Brodegard. Just a Fad? Gamification in Health and Fitness Apps. *JMIR Serious Games*, 2(2), 2014.

[LWM+11] Marcello La Rosa, Petia Wohed, Jan Mendling, Arthur H. M. ter Hofstede, Hajo A. Reijers, and Wil M. P. van der Aalst. Managing Process Model Complexity Via Abstract Syntax Modifications. *IEEE Transactions on Industrial Informatics*, 7(4):614–629, 2011.

[LY92] Gang Soo Lee and Jung-Mo Yoon. An Empirical Study on the Complexity Metrics of Petri Nets. *Microelectronics Reliability*, 32(3):323–329, 1992.

[Mac88] Jock D. Mackinlay. Applying a Theory of Graphical Rresentation to the Graphic Design of User Interfaces. In *Proceedings of the 1st ACM SIGGRAPH Symposium on User Interface Software (UIST 1988)*, pages 179–189, Alberta, Canada, 1988.

[Mah36] Prasanta C. Mahalanobis. On the Generalized Distance in Statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[Mal80] Thomas W. Malone. What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games. In *Proceedings of the 3rd ACM Symposium on Small Systems (SIGSMALL 1980)*, pages 162–169, Palo Alto, CA, United States, 1980.

[Mal82] Thomas W. Malone. Heuristics for Designing Enjoyable User Interfaces: Lessons From Computer Games. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems (CHI 1982)*, pages 63–68, Gaithersburg, MD, United States, 1982.

[Mar15] Andrzej Marczewski. *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design.* CreateSpace Independent Publishing Platform, North Charleston, SC, United States, 1st edition, 2015.

[Maz09] Riccardo Mazza. *Introduction to Information Visualization.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2009.

[MBOT13a] Elisa D. Mekler, Florian Brühlmann, Klaus Opwis, and Alexandre N. Tuch. Disassembling Gamification: The Effects of Points and Meaning on User Motivation and Performance. In *Proceedings of the 2013 Conference on Human Factors in Computing Systems (CHI 2013)*, pages 1137–1142, Paris, France, 2013.

[MBOT13b] Elisa D. Mekler, Florian Brühlmann, Klaus Opwis, and Alexandre N. Tuch. Do Points, Levels and Leaderboards Harm Intrinsic Motivation? An Empirical Analysis of Common Gamification Elements. In *Proceedings of the 1st International Conference on Gameful Design, Research, and Applications (Gamification 2013)*, pages 66–73, Stratford, Canada, 2013.

[MBTO15] Elisa D. Mekler, Florian Brühlmann, Alexandre N. Tuch, and Klaus Opwis. Towards Understanding the Effects of Individual Gamification Elements on Intrinsic Motivation and Performance. *Computers in Human Behavior*, 71:525–534, 2015.

[MC05] David Michael and Sande Chen. *Serious Games: Games that Educate, Train, and Inform.* Thomson Course Technology, Boston, MA, United States, 1st edition, 2005.

[McC76] Thomas J. McCabe. A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2(4):308–320, 1976.

[McC92] S. McCready. There is More Than One Kind of Workflow Software. *Computerworld*, 2(November):86–90, 1992.

[McG11] Jane McGonigal. *Reality is Broken: Why Games Make Us Better and How They Can Change the World.* The Penguin Press, New York City, NY, United States, 1st edition, 2011.

[McL69] G. Harry McLaughlin. SMOG Grading: A New Readability Formula. *Journal of Reading*, 12(8):639–646, 1969.

[MCMN05] Carsten Magerkurth, Adrian David Cheok, Regan L. Mandryk, and Trond Nilsen. Pervasive Games: Bringing Computer Entertainment Back to the Real World. *ACM Computers in Entertainment*, 3(3):1–19, 2005.

[Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2008.

[Men09] Jan Mendling. Empirical Studies in Process Model Verification. In Kurt Jensen and Wil M. P. van der Aalst, editors, *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems*, chapter 12, pages 208–224. Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2009.

[MHK16] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. Gamification in Crowdsourcing: A Review. In *Proceedings of the 49th Annual Hawaii International Conference on System Sciences (HICSS 2016)*, pages 4375–4384, Koloa, HI, United States, 2016.

[Min01] John Mingers. Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research*, 12(3):240–259, 2001.

[MJ10] M. Lynne Markus and Dax D. Jacobson. Business Process Governance. In Jan vom Brocke and Michael Rosemann, editors, *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture*, chapter 10, pages 201–222. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[MK14] Tobias Mutter and Dennis Kundisch. Behavioral Mechanisms Prompted by Badges: The Goal-Gradient Hypothesis. In *Proceedings of the 35th International Conference on Information Systems (ICIS 2014)*, Auckland, New Zealand, 2014.

[MK15] Tobias Mutter and Dennis Kundisch. Behavioral Mechanisms Prompted By Virtual Rewards: The Small-Area Hypothesis. In *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*, Münster, Germany, 2015.

[MMN+06] Jan Mendling, M. Moser, Gustaf Neumann, H. M. W. Verbeek, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. Faulty EPCs in the SAP Reference Model. In *Proceedings of the 4th International Conference on Business Process Management (BPM 2006)*, pages 451–457, Vienna, Austria, 2006.

[MMWFF92] Raúl Medina-Mores, Terry Winograd, Rodrigo Flores, and Fernando Flores. The Action Workflow Approach to Workflow Management Technology. In *Proceedings of the 4th ACM Conference on Computer-Supported Cooperative Work (CSCW 1992)*, pages 281–288, Toronto, Canada, 1992.

[MNvdA07] Jan Mendling, Gustaf Neumann, and Wil M. P. van der Aalst. Understanding the Occurrence of Errors in Process Models Based on Metrics. In *Proceedings of the 2007 International Conference on Cooperative Information Systems (CoopIS 2007)*, pages 113–130, Vilamoura, Portugal, 2007.

[Mon05] Markus Montola. Exploring the Edge of the Magic Circle: Defining Pervasive Games. In *Proceedings of the 6th Digital Arts and Culture Conference (DAC 2005)*, Copenhagen, Denmark, 2005.

[Moo05] Daniel L. Moody. Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.

[Moo09] Daniel L. Moody. The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.

[Mor78] Jacob L. Moreno. *Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama.* Beacon House, Beacon, NY, United States, 3rd edition, 1978.

[Mor99] Sandro Morasca. Measuring Attributes of Concurrent Software Specifications in Petri Nets. In *Proceedings of the 6th International Symposium on Software Metrics (METRICS 1999)*, pages 100–110, Boca Raton, FL, United States, 1999.

[MPG08] Kamal Masri, Drew Parker, and Andrew Gemino. Using Iconic Graphics in Entity-Relationship Diagrams: The Impact on Understanding. *Journal of Database Management*, 19(3):22–41, 2008.

[MPP11] Scott B. Mackenzie, Philip M. Podsakoff, and Nathan P. Podsakoff. Construct Measurement and Validation Procedures in MIS and Behavioral Research: Integrating New and Existing Techniques. *MIS Quarterly*, 35(2):293–334, 2011.

[MR07] Jan Mendling and Jan C. Recker. Extending the Discussion of Model Quality: Why Clarity and Completeness May Not Always Be Enough. In *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007)*, pages 109–121, Trondheim, Norway, 2007.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In *Proceedings of the 5th International Conference on Business Process Management (BPM 2007)*, pages 48–63, Brisbane, Australia, 2007.

[MRGAM15] Alberto Mora, Daniel Riera, Carina González, and Joan Arnedo-Moreno. A Literature Review of Gamification Design Frameworks. In *Proceedings of the 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games 2015)*, pages 1–8, Skövde, Sweden, 2015.

[MRR09] Jan Mendling, Jan C. Recker, and Hajo A. Reijers. Process Modeling Quality: A Framework and Research Agenda. In *BPM Reports (0902)*, BPM Center, Eindhoven, Netherlands, 2009.

[MRR10a] Jan Mendling, Jan C. Recker, and Hajo A. Reijers. On the Usage of Labels and Icons in Business Process Modeling. *International Journal of Information System Modeling and Design*, 1(2):40–58, 2010.

[MRR10b] Jan Mendling, Hajo A. Reijers, and Jan C. Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems*, 35(4):467–482, 2010.

[MRvdA10] Jan Mendling, Hajo A. Reijers, and Wil M. P. van der Aalst. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology*, 52(2):127–136, 2010.

[MRW12] Jan Mendling, Jan C. Recker, and Johannes Wolf. Collaboration Features in Current BPM Tools. *EMISA Forum*, 32(1):48–65, 2012.

[MS95] Salvatore T. March and Gerald F. Smith. Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4):251–266, 1995.

[MS08b] Jan Mendling and Mark Strembeck. Influence Factors of Understanding Business Process Models. In *Proceedings of the 11th International Conference on Business Information Systems (BIS 2008)*, pages 142–153, Innsbruck, Austria, 2008.

[MS14] Isel Moreno-Montes de Oca and Monique Snoeck. Pragmatic Guidelines for Business Process Modeling. Technical report, KU Leuven, Faculty of Economics and Business, Leuven, Netherlands, 2014.

[MSBS02] Daniel L. Moody, Guttorm Sindre, Terje Brasethvik, and Arne Sølvberg. Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework. In *Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002)*, pages 380–396, Tampere, Finland, 2002.

[MSGGL12] Jan Mendling, Laura Sanchez-Gonzalez, Felix Garcia, and Marcello La Rosa. Thresholds for Error Probability Measures of Business Process Models. *Journal of Systems and Software*, 85(5):1188–1197, 2012.

[MSR12] Jan Mendling, Mark Strembeck, and Jan C. Recker. Factors of Process Model Comprehension - Findings from a Series of Experiments. *Decision Support Systems*, 53(1):195–206, 2012.

[MSW09] Markus Montola, Jaakko Stenros, and Annika Waern. *Pervasive Games: Theory and Design.* Morgan Kaufmann, San Francisco, CA, United States, 1st edition, 2009.

[MTJ$^+$10] Hafedh Mili, Guy Tremblay, Guitta Bou Jaoude, Éric Lefebvre, Lamia Elabed, and Ghizlane El Boussaidi. Business Process Modeling Languages: Sorting Through the Alphabet Soup. *ACM Computing Surveys*, 43(1), 2010.

[Mur89] Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[MVvD$^+$08] Jan Mendling, H. M. W. Verbeek, Boudewijn F. van Dongen, Wil M. P. van der Aalst, and Gustaf Neumann. Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering*, 64(1):312–329, 2008.

[MW47] Henry B. Mann and Donald R. Whitney. On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[MWBR12] Paul Mathiesen, Jason Watson, Wasana Bandara, and Michael Rosemann. Applying Social Technology to Business Process Lifecycle Management. In *Proceedings of the 4th Workshop on Business Process Management and Social Software (BPMS2 2011)*, pages 231–241, Clermont-Ferrand, France, 2012.

[MWHA17] Benedikt Morschheuser, Karl Werder, Juho Hamari, and Julian Abe. How to Gamify? A Method for Designing Gamification. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS 2017)*, pages 1298–1307, Waikoloa, HI, United States, 2017.

[NC02] Jeanne Nakamura and Mihály Csíkszentmihályi. The Concept of Flow. In Charles R. Snyder and Shane J. Lopez, editors, *Handbook of Positive Psychology*, chapter 7, pages 89–105. Oxford University Press, Oxford, United Kingdom, 1st edition, 2002.

[NE09] Gustaf Neumann and Selim Erol. From a Social Wiki to a Social Workflow System. In *Proceedings of the 1st Workshop on Business Process Management and Social Software (BPMS2 2008)*, pages 698–708, Milano, Italy, 2009.

[Nel12] Mark J. Nelson. Soviet and American Precursors to the Gamification of Work. In *Proceedings of the 16th International Academic MindTrek Conference (MindTrek 2012)*, pages 23–26, Tampere, Finland, 2012.

[Neu09] Thomas Neubauer. An Empirical Study About the Status of Business Process Management. *Business Process Management Journal*, 15(2):166–183, 2009.

[Neu14] W. Lawrence Neuman. *Social Research Methods: Qualitative and Quantitative Approaches.* Pearson Education, Essex, United Kingdom, 7th edition, 2014.

[nG16] newzoo Games. 2016 Global Games Market Report. Technical report, newzoo Games, 2016.

[NH11] Björn Niehaves and Jörn Henser. Boundary Spanning Practices in BPM: A Dynamic Capability Perspective. In *Proceedings of the 17th Americas Conference on Information Systems (AMCIS 2011)*, Detroit, MI, United States, 2011.

[Nis98] Mark E. Nissen. Redesigning Reengineering through Measurement. *MIS Quarterly*, 22(4):509–534, 1998.

[Nor50] Fritz Nordsieck. *Die schaubildliche Erfassung und Untersuchung der Betriebsorganisation.* C. E. Poeschel Verlag, Stuttgart, Germany, 4th edition, 1950.

[NP11] Björn Niehaves and Ralf Plattfaut. Collaborative Business Process Management: Status Quo and Quo Vadis. *Business Process Management Journal*, 17(3):384–402, 2011.

[NP12] Alexander Nolte and Michael Prilla. Normal Users Cooperating on Process Models: Is It Possible at All? In *Proceedings of the 18th International Workshop on Groupware (CRIWG 2012)*, pages 57–72, Raesfeld, Germany, 2012.

[NPGP12] H. James Nelson, Geert Poels, Marcela Genero, and Mario Piattini. A Conceptual Modeling Quality Framework. *Software Quality Journal*, 20(1):201–228, 2012.

[NZT+14] Fiona Fui-Hoon Nah, Qing Zeng, Venkata Rajasekhar Telaprolu, Abhishek Padmanabhuni Ayyappa, and Brenda Eschenbrenner. Gamification of Education: A Review of Literature. In *Proceedings of the 1st International Conference on HCI in Business (HCIB 2014)*, pages 401–409, Heraklion, Greece, 2014.

[OA11] Azmi Omar and Nazlena Mohamad Ali. Measuring Flow in Gaming Platforms. In *Proceedings of the 2011 International Conference on Semantic Technology and Information Retrieval (STAIR 2011)*, pages 302–305, Putrajaya, Malaysia, 2011.

[OB91] Wanda J. Orlikowski and Jack J. Baroudi. Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, 2(1):1–28, 1991.

[OBS12] Sven Overhage, Dominik Q. Birkmeier, and Sebastian Schlauderer. Quality Marks, Metrics, and Measurement Procedures for Business Process Models: The 3QM-Framework. *Business & Information Systems Engineering*, 4(5):229–246, 2012.

[OFR+12] Avner Ottensooser, Alan Fekete, Hajo A. Reijers, Jan Mendling, and Con Menictas. Making Sense of Business Process Descriptions: An Experimental Comparison of Graphical and Textual Notations. *Journal of Systems and Software*, 85(3):596–606, 2012.

[OJK14] Florin Oprescu, Christian Jones, and Mary Katsikitis. I PLAY AT WORK—Ten Principles for Transforming Work Processes through Gamification. *Frontiers in Psychology*, 5:1–5, 2014.

[OS10] Chitu Okoli and Kira Schabram. A Guide to Conducting a Systematic Literature Review of Information Systems Research. *Working Papers on Information Systems*, 10(26):1–51, 2010.

[Oul95] Martyn A. Ould. *Business Processes: Modelling and Analysis for Re-Engineering and Improvement.* Wiley, Hoboken, NJ, United States, 1995.

[Pag13] Robert R. Pagano. *Understanding Statistics in the Behavioral Sciences.* Wadsworth Cengage Learning, Belmont, CA, United States, 10th edition, 2013.

[Pai86] Allan Paivio. *Mental Representations: A Dual Coding Approach.* Oxford University Press, Oxford, United Kingdom, 1986.

[PBD+13] Nicolas Pflanzl, Dominic Breuker, Hanns-Alexander Dietrich, Matthias Steinhorst, Maria Shitkova, Jörg Becker, and Patrick Delfmann. A Framework for Fast Graph-Based Pattern Matching in Conceptual Models. In *Proceedings of the 15th IEEE Conference on Business Informatics (CBI 2013)*, pages 250–257, Vienna, Austria, 2013.

[PBSV15a] Nicolas Pflanzl, Katrin Bergener, Armin Stein, and Gottfried Vossen. Information Systems Freshmen Teaching: Case Experience from Day One. *International Journal of Information and Operations Management Education*, 6(1):49–69, 2015.

[PBSV15b] Nicolas Pflanzl, Katrin Bergener, Armin Stein, and Gottfried Vossen. Information Systems Freshmen Teaching: Case Experience from Day One. 2015.

[PCJ95] Helen C. Purchase, Robert F. Cohen, and Murray I. James. Validating Graph Drawing Aesthetics. In *Proceedings of the 4th International Symposium on Graph Drawing (1995)*, pages 435–446, Passau, Germany, 1995.

[PCJ97] Helen C. Purchase, Robert F. Cohen, and Murray I. James. An Experimental Study of the Basis for Graph Drawing Algorithms. *Journal of Experimental Algorithmics*, 2(4):1–17, 1997.

[Pea84] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Boston, MA, United States, 1st edition, 1984.

[Pet62] Carl Adam Petri. *Kommunikation mit Automaten.* Doctoral dissertation, Technische Hochschule Darmstadt, 1962.

[Pet94] Robert A. Peterson. A Meta-analysis of Cronbach's Coefficient Alpha. *Journal of Consumer Research*, 21(2):381–391, 1994.

[Pet95] Marian Petre. Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. *Communications of the ACM*, 38(6):33–44, 1995.

[Pet06] Marian Petre. Cognitive Dimensions 'Beyond the Notation'. *Journal of Visual Languages and Computing*, 17(4):292–301, 2006.

[Pfl16] Nicolas Pflanzl. Gameful Business Process Modeling. In *Proceedings of the 7th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA 2016)*, pages 17–20, Vienna, Austria, 2016.

[PGBP15] Oscar Pedreira, Félix García, Nieves Brisaboa, and Mario Piattini. Gamification in Software Engineering - A Systematic Mapping. *Information and Software Technology*, 57:157–168, 2015.

[Pin11] Daniel H. Pink. *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books, New York City, NY, United States, 1st edition, 2011.

[PK93] Alain Pinsonneault and Kenneth L. Kraemer. Survey Research Methodology in Management Information Systems: An Assessment. *Journal of Management Information Systems*, 10(2):75–105, 1993.

[PLM15] Fabian Pittke, Henrik Leopold, and Jan Mendling. Automatic Detection and Resolution of Lexical Ambiguity in Process Models. *IEEE Transactions on Software Engineering*, 41(6):526–544, 2015.

[PLX07] Paul A. Pavlou, Huigang Liang, and Yajiong Xue. Understanding and Mitigating Uncertainty in Online Exchnge Relationships: A Principle-Agent Perspective. *MIS Quarterly*, 31(1):105–136, 2007.

[PM85] Michael E. Porter and Victor E. Millar. How Information Gives You Competitive Advantage. *Harvard Buiness Review*, 63(4):149–160, 1985.

[PMCC01] Helen C. Purchase, Matthew McGill, Linda Colpoys, and David Carrington. Graph Drawing Aesthetics and the Comprehension of UML Class Diagrams: An Empirical Study. In *Proceedings of the 1st Asia-Pacific Symposium on Information Visualisation (APVis 2001)*, pages 129–137, Sydney, Australia, 2001.

[Por85] Michael E. Porter. *Competitive Advantage - Creating and Sustaining Superior Performance*. The Free Press, New York City, NY, United States, 1st edition, 1985.

[Pow95] Thomas C. Powell. Total Quality Management as Competitive Advantage: A Review and Empirical Study. *Strategic Management Journal*, 16(1):15–37, 1995.

[PP91] Elazar J. Pedhazur and Liora Pedhazur Schmelkin. *Measurement, Design, and Analysis: An Integrated Approach*. Lawrence Erlbaum Associates, Hillsdale, NJ, United States, 1st edition, 1991.

[PPP12] Helen C. Purchase, Christopher Pilcher, and Beryl Plimmer. Graph Drawing Aesthetics—Created by Users, Not Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):81–92, 2012.

[Pre01] Marc Prensky. *Digital Game-Based Learning*. McGraw-Hill, New York City, NY, United States, 1st edition, 2001.

[Pre03] Marc Prensky. Digital Game-based Learning. *ACM Computers in Entertainment*, 1(1):1–21, 2003.

[Pri10] Michael Prilla. Models, Social Tagging and Knowledge Management – A fruitful Combination for Process Improvement. In *Proceedings of the 2nd Workshop on Business Process Management and Social Software (BPMS2 2009)*, pages 266–277, Ulm, Germany, 2010.

[PRR10] Andrew K. Przybylski, C. Scott Rigby, and Richard M. Ryan. A Motivational Model of Video Game Engagement. *Review of General Psychology*, 14(2):154–166, 2010.

[PRR13] Stephan Poelmans, Hajo A. Reijers, and Jan C. Recker. Investigating the Success of Obusiness Process Management Systems. *Information Technology and Management*, 14(4):295–314, 2013.

[PT00] A. Papakostas and Ioannis G. Tollis. Efficient Orthogonal Drawings of High Degree Graphs. *Algorithmica*, 26(1):100–125, 2000.

[PT15] Lisa-Maria Putz and Horst Treiblmaier. Creating a Theory-Based Research Agenda for Gamification. In *Proceedings of the 20th Americas Conference on Information Systems (AMCIS 2015)*, pages 1–13, Puerto Rico, PR, United States, 2015.

[PTC13] Pablo Paredes, Anuj Tewari, and John Canny. Design Principles for the Conceptualization of Games for Health Behavior Change. In *Proceedings of the CHI 2013 Workshop on Designing Gamification: Creating Gameful and Playful Experiences*, Paris, France, 2013.

[PTRC07] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3):45–77, 2007.

[Puc11] Max J. Pucher. How to Link BPM Governance and Social Collaboration Through an Adaptive Paradigm. In Layna Fischer, editor, *Social BPM - Work, Planning and Collaboration Under the Impact of Social Technology*, chapter 6, pages 57–76. Future Strategies Inc., Lighthouse Point, FL, United States, 1st edition, 2011.

[Pur97] Helen C. Purchase. Which Aesthetic Has the Greatest Effect on Human Understanding? In *Proceedings of the 5th International Symposium on Graph Drawing (GD 1997)*, pages 248–261, Rome, Italy, 1997.

[Pur00] Helen C. Purchase. Effective Information Visualisation: A Study of Graph Drawing Aesthetics and Algorithms. *Interacting with Computers*, 13(2):147–162, 2000.

[Pur02a] Helen C. Purchase. Empirical Evaluation of Aesthetics-Based Graph Layout. *Empirical Software Engineering*, 7(3):233–255, 2002.

[Pur02b] Helen C. Purchase. Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, 2002.

[Pur14] Helen C. Purchase. Twelve Years of Diagrams Research. *Journal of Visual Languages and Computing*, 25(2):57–75, 2014.

[PV13b] Nicolas Pflanzl and Gottfried Vossen. Human-Oriented Challenges of Social BPM: An Overview. In *Proceedings of the 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2013)*, pages 163–176, St. Gallen, Switzerland, 2013.

[PV14] Nicolas Pflanzl and Gottfried Vossen. Challenges of Social Business Process Management. In *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS 2014)*, pages 3868–3877, Waikoloa, HI, United States, 2014.

[PWS08] David Pinelle, Nelson Wong, and Tadeusz Stach. Heuristic Evaluation for Games: Usability Principles for Video Game Design. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems (CHI 2008)*, pages 1453–1462, Florence, Italy, 2008.

[PZW+12] Jakob Pinggera, Stefan Zugal, Matthias Weidlich, Dirk Fahland, Barbara Weber, Jan Mendling, and Hajo A. Reijers. Tracing the Process of Process Modeling with Modeling Phase Diagrams. In *Proceedings of the 2nd International Workshop on Empirical Research in Business Process Management (ER-BPM 2011)*, pages 370–382, Clermont-Ferrand, France, 2012.

[QB11] Alexander J. Quinn and Benjamin B. Bederson. Human Computation: A Survey and Taxonomy of a Growing Field. In *Proceedings of the 2011 International Conference on Human Factors in Computing Systems (CHI 2011)*, pages 1403–1412, Vancouver, Canada, 2011.

[Rac14] Felix Raczkowski. Making Points the Point: Towards a History of Ideas of Gamification. In Mathias Fuchs, Sonia Fizek, Paolo Ruffino, and Niklas Schrape, editors, *Rethinking Gamification*, chapter 7, pages 141–160. meson press, Lüneburg, Germany, 2014.

[Rae06] Joost Raessens. Playful Identities, or the Ludification of Culture. *Games and Culture*, 1(1):52–57, 2006.

[RB10] Pascal Ravesteyn and Ronald Batenburg. Surveying the Critical Success Factors of BPM-Systems Implementation. *Business Process Management Journal*, 16(3):492–507, 2010.

[RB13a] David Robinson and Victoria Bellotti. A Preliminary Taxonomy of Gamification Elements for Varying Anticipated Commitment. In *Proceedings of the*

CHI 2013 Workshop on Designing Gamification: Creating Gameful and Playful Experiences, Paris, France, 2013.

[RB13b] Geary A. Rummler and Alan P. Brache. *Improving Performance: How to Manage the White Space on the Organization Chart.* Jossey-Bass, San Francisco, CA, United States, 3rd edition, 2013.

[RD00a] Richard M. Ryan and Edward L. Deci. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1):54–67, jan 2000.

[RD00b] Richard M. Ryan and Edward L. Deci. Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist*, 55(1):68–78, 2000.

[RD00c] Richard M. Ryan and Edward L. Deci. When Rewards Compete with Nature: The Undermining of Intrinsic Motivation and Self-Regulation. In Carol Sansone and Judith Harackiewicz, editors, *Intrinsic and Extrinsic Motivation: The Search for Optimal Motivation and Performance*, chapter 2, pages 13–54. Academic Press, San Diego, CA, United States, 1st edition, 2000.

[Rec12a] Jan C. Recker. Empirical Investigation of the Usefulness of Gateway Constructs in Process Models. *European Journal of Information Systems*, 22(6):673–689, 2012.

[Rec12b] Jan C. Recker. "Modeling With Tools is Easier, Believe Me" — The Effects of Tool Functionality on Modeling Grammar Usage Beliefs. *Information Systems*, 37(3):213–226, 2012.

[Rec13] Jan C. Recker. *Scientific Research in Information Systems.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2013.

[Rei82] Wolfgang Reisig. *Petrinetze: Eine Einführung.* Springer, Berlin, Heidelberg, Germany, 1st edition, 1982.

[RFME11] Hajo A. Reijers, Thomas Freytag, Jan Mendling, and Andreas Eckleder. Syntax Highlighting in Business Process Models. *Decision Support Systems*, 51(3):339–349, 2011.

[Ric11] Clay Richardson. Foreword. In Layna Fischer, editor, *Social BPM - Work, Planning and Collaboration Under the Impact of Social Technology*, chapter 1, pages 5–8. Future Strategies Inc., Lighthouse Point, FL, United States, 1st edition, 2011.

[Rit10a] Peter Rittgen. Quality and Perceived Usefulness of Process Models. In *Proceedings of the 25th ACM Symposium on Applied Computing (SAC 2010)*, pages 65–72, Sierre, Switzerland, 2010.

[Rit10b] Peter Rittgen. Success Factors of e-Collaboration in Business Process Modeling. In *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE 2010)*, pages 24–37, Hammamet, Tunisia, 2010.

[RL05] Hajo A. Reijers and S. Liman Mansar. Best Practices in Business Process Redesign: An Overview and Qualitative Evaluation of Successful Redesign Heuristics. *Omega*, 33(4):283–306, 2005.

[RM08] Hajo A. Reijers and Jan Mendling. Modularity in Process Models: Review and Effects. In *Proceedings of the 6th International Conference on Business Process Management (BPM 2008)*, pages 20–35, Milan, Italy, 2008.

[RM11] Hajo A. Reijers and Jan Mendling. A Study Into the Factors That Influence the Understandability of Business Process Models. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(3):449–462, 2011.

[RMH13] Jan C. Recker, Jan Mendling, and Christopher Hahn. How Collaborative Ttechnology Supports Cognitive Processes in Collaborative Process Modeling: A Capabilities-Gains-Outcome Model. *Information Systems*, 38(8):1031–1045, 2013.

[RMR10] Hajo A. Reijers, Jan Mendling, and Jan C. Recker. Business Process Quality Management. In Jan vom Brocke and Michael Rosemann, editors, *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, chapter 8, pages 167–185. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[Ros03] Michael Rosemann. Preparation of Process Modeling. In Jörg Becker, Martin Kugeler, and Michael Rosemann, editors, *Process Management: A Guide for the Design of Business Processes*, chapter 3, pages 41–78. Springer, Berlin, Heidelberg, Germany, 2003.

[Ros06a] Michael Rosemann. Potential Pitfalls of Process Modeling: Part A. *Business Process Management Journal*, 12(2):249–254, 2006.

[Ros06b] Michael Rosemann. Potential Pitfalls of Process Modeling: Part B. *Business Process Management Journal*, 12(3):377–384, 2006.

[Rou05] Richard Rouse. *Game Design: Theory and Practice*. Wordware Publishing, Plano, TX, United States, 2nd edition, 2005.

[RR98a] Wolfgang Reisig and Grzegorz Rozenberg. *Lectures on Petri Nets I: Basic Models*. Springer, Berlin, Heidelberg, Germany, 1st edition, 1998.

[RR98b] Wolfgang Reisig and Grzegorz Rozenberg. *Lectures on Petri Nets II: Applications*. Springer, Berlin, Heidelberg, Germany, 1st edition, 1998.

[RRIG09] Jan C. Recker, Michael Rosemann, Marta Indulska, and Peter Green. Business Process Modeling: A Comparative Analysis. *Journal of the Association for Information Systems*, 10(4):333–363, 2009.

[RRP06] Richard M. Ryan, C. Scott Rigby, and Andrew K. Przybylski. The Motivational Pull of Video Games: A Self-Determination Theory Approach. *Motivation and Emotion*, 30(4):344–360, 2006.

[RRvdW14] Jan C. Recker, Hajo A. Reijers, and Sander G. van de Wouw. Process Model Comprehension: The Effects of Cognitive Abilities, Learning Style, and Strategy. *Communications of the Association for Information Systems*, 34(9):199–222, 2014.

[RS15] Suman Roy and A. S. M. Sajeev. A Formal Framework for Diagnostic Analysis for Errors of Business Processes. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9410, pages 226–261, 2015.

[RSR10] Jan C. Recker, Niz Safrudin, and Michael Rosemann. How Novices Model Business Processes. In *Proceedings of the 8th International Conference on Business Process Management (BPM 2010)*, pages 29–44, Hoboken, NJ, United States, 2010.

[RSR12] Jan C. Recker, Norizan Safrudin, and Michael Rosemann. How Novices Design Business Processes. *Information Systems*, 37(6):557–573, 2012.

[RSS01] Michael Rosemann, Wasana Sedera, and Darshana Sedera. Testing a Framework for the Quality of Process Models - A Case Study. In *Proceedings of the 5th Pacific Asia Conference on Information Systems (PACIS 2001)*, pages 978–991, Seoul, South Korea, 2001.

[RTT16] Maximilian Riefer, Simon Felix Ternis, and Tom Thaler. Mining Process Models from Natural Language Text: A State-of-the-Art Analysis. In *Proceedings of the 2016 Multikonferenz Wirtschaftsinformatik (MKWI 2016)*, Ilmenau, Germany, 2016.

[Rub87] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons, Hoboken, NJ, United States, 1st edition, 1987.

[Rus11] Steve Russell. How Social Technologies Enhance the BPM Experience for All Participants. In Layna Fischer, editor, *Social BPM - Work, Planning and Collaboration Under the Impact of Social Technology*, chapter 10, pages 113–122. Future Strategies Inc., Lighthouse Point, FL, United States, 1st edition, 2011.

[RvB10] Michael Rosemann and Jan vom Brocke. The Six Core Elements of Business Process Management. In Jan vom Brocke and Michael Rosemann, editors, *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, chapter 5, pages 107–122. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[RVE03] Falko Rheinberg, Regina Vollmeyer, and Stefan Engeser. Die Erfassung des Flow-Erlebens. In Joachim Stiensmeier-Pelster and Falko Rheinberg, editors, *Diagnostik von Motivation und Selbstkonzept*, chapter 14, pages 261–280. Hogrefe, Göttingen, Germany, 1st edition, 2003.

[RvWML10] Hajo A. Reijers, Sander van Wijk, Bela Mutschler, and Maarten Leurs. BPM in Practice: Who Is Doing What? In *Proceedings of the 8th International Conference on Business Process Management (BPM 2010)*, pages 45–60, Hoboken, NJ, United States, 2010.

[RW82] Bernd Rosenstengel and Udo Winand. *Petri-Netze: Eine anwendungsorientierte Einführung*. Friedr. Vieweg & Sohn, Braunschweig, Germany, 1st edition, 1982.

[RZSL06] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The Value of Reputation on eBay: A Controlled Experiment. *Experimental Economics*, 9(2):79–101, 2006.

[SA14] Christian Schlagenhaufer and Michael Amberg. Psychology Theories in Gamification: A Review of Information Systems Literature. In *Proceedings of the 11th European, Mediterranean and Middle Eastern Conference on Information Systems (EMCIS 2014)*, pages 1–12, Doha, Qatar, 2014.

[SA15] Christian Schlagenhaufer and Michael Amberg. A Descriptive Literature Review and Classification Framework for Gamification in Information Systems. In *Proceedings of the 23rd European Conference on Information Systems (ECIS 2015)*, Münster, Germany, 2015.

[SAK15] Frank Schoenthaler, Dominik Augenstein, and Thomas Karle. Design and Governance of Collaborative Business Processes in Industry 4.0. In *Proceedings of the 2015 Workshop on Cross-Organizational and Cross-Company BPM (XOC-BPM 2015)*, Lisbon, Portugal, 2015.

[Sar15] Advait Sarkar. The Impact of Syntax Colouring on Program Comprehension. In *Proceedings of the 26th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2015)*, pages 49–58, Bournemouth, United Kingdom, 2015.

[SB10] August-Wilhelm Scheer and Eric Brabänder. The Process of Business Process Management. In Jan vom Brocke and Michael Rosemann, editors, *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture*, chapter 12, pages 239–267. Springer, Berlin, Heidelberg, Germany, 1st edition, 2010.

[SBG04] Detmar Straub, Marie-Claude Boudreau, and David Gefen. Validation Guidelines for IS Positivist Research. *Communications of the Association for Information Systems*, 13(24):380–427, 2004.

[Sch00] August-Wilhelm Scheer. *ARIS - Business Process Modeling*. Springer, Berlin, Heidelberg, Germany, 3rd edition, 2000.

[Sch08] Jesse Schell. *The Art of Game Design: A Book of Lenses*. Morgan Kaufmann Publishers, Burlington, MA, United States, 1st edition, 2008.

[SDF+04] Sherry Shavor, Jim D'Anjou, Scott Fairbrother, Dan Kehn, John Kellerman, and Pat McCarthy. *Eclipse: Anwendungen und Plug-Ins mit Java entwickeln*. Addison-Wesley, Munich, Germany, 1st edition, 2004.

[SF15b] Katie Seaborn and Deborah I. Fels. Gamification in Theory and Action: A Survey. *International Journal of Human-Computer Studies*, 74:14–31, 2015.

[SGGRM12] Laura Sánchez-González, Félix García, Francisco Ruiz, and Jan Mendling. Quality Indicators for Business Process Models from a Gateway Complexity Perspective. *Information and Software Technology*, 54(11):1159–1174, 2012.

[Sha48] Claude Edmund Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[Sim96] Herbert A. Simon. *The Sciences of the Artificial*. The MIT Press, Cambridge, MA, United States, 3rd edition, 1996.

[SJ09] Roodt Sumarie and Pieter Joubert. Evaluating Serious Games in Higher Education: A Theory-based Evaluation of IBMs Innov8. In *Proceedings of the 3rd European Conference on Games Based Learning (ECGBL 2009)*, pages 332–338, Graz, Austria, 2009.

[SJW12] Penelope Sweetser, Daniel M. Johnson, and Peta Wyeth. Revisiting the GameFlow Model with Detailed Heuristics. *Journal of Creative Technologies*, 3, 2012.

[SK09] August-Wilhelm Scheer and Joerg Klueckmann. BPM 3.0. In *Proceedings of the 7th International Conference on Business Process Management (BPM 2009)*, pages 15–27, Ulm, Germany, 2009.

[SKW12] Pnina Soffer, Maya Kaner, and Yair Wand. Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In *Proceedings of the 2nd International Workshop on Empirical Research in Business Process Management (ER-BPM 2011)*, pages 357–369, Clermont-Ferrand, France, 2012.

[Smi76] Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. W. Strahan and T. Cadell, London, United Kingdom, 1776.

[SMM+10] António Rito Silva, Rachid Meziani, Rodrigo Magalhães, David Martinho, Ademar Aguiar, and Nuno Flores. AGILIPO: Embedding Social Software Features into Business Process Tools. In *Proceedings of the 2nd Workshop on Business Process Management and Social Software (BPMS2 2009)*, pages 219–230, Ulm, Germany, 2010.

[SMSZ13] Pol Schumacher, Mirjam Minor, and Eric Schulte-Zurhausen. Extracting and Enriching Workflows from Text. In *Proceedings of the 14th IEEE International Conference on Information Reuse and Integration (IRI 2013)*, pages 285–292, San Francisco, CA, United States, 2013.

[SN09] Rainer Schmidt and Selmin Nurcan. BPM and Social Software. In *Proceedings of the 1st Workshop on Business Process Management and Social Software (BPMS2 2008)*, pages 649–658, Milano, Italy, 2009.

[SN10] Rainer Schmidt and Selmin Nurcan. Augmenting BPM with Social Software. In *Proceedings of the 2nd Workshop on Business Process Management and Social Software (BPMS2 2009)*, pages 201–206, Ulm, Germany, 2010.

[SR98] Reinhard Schuette and Thomas Rotthowe. The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER 1998)*, pages 240–254, Singapore, 1998.

[SR12] António Rito Silva and Michael Rosemann. Processpedia: An Ecological Environment for BPM Stakeholders' Collaboration. *Business Process Management Journal*, 18(1):20–42, 2012.

[SS77] John Miles Smith and Diane C. P. Smith. Database Abstractions: Aggregation and Generalization. *ACM Transactions on Database Systems*, 2(2):105–133, 1977.

[SSHJ12] Jim Sinur, W. Roy Schulte, Janelle B. Hill, and Teresa Jones. Magic Quadrant for Intelligent Business Process Management Suites. 2012.

[ST13] Rok Skrinjar and Peter Trkman. Increasing Process Orientation with Business Process Management: Critical Practices. *International Journal of Information Management*, 33(1):48–60, 2013.

[Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Springer, Vienna, Austria, 1st edition, 1973.

[STD12] Ognjen Scekic, Hong-Linh Truong, and Schahram Dustdar. Modeling Rewards and Incentive Mechanisms for Social BPM. In *Proceedings of the 10th International Conference on Business Process Management (BPM 2012)*, pages 150–155, Tallinn, Estonia, 2012.

[Sur05] James Surowiecki. *The Wisdom of Crowds.* Anchor, New York City, NY, United States, 1st edition, 2005.

[SvBR13] Theresa Schmiedel, Jan vom Brocke, and Jan C. Recker. Which Cultural Values Matter to Business Process Management? Results from a Global Delphi Study. *Business Process Management Journal*, 19(2):292–317, 2013.

[SVOK12] Frank Schönthaler, Gottfried Vossen, Andreas Oberweis, and Thomas Karle. *Business Processes for Business Communities: Modeling Languages, Methods, Tools.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2012.

[SW65] Samuel S. Shapiro and Martin B. Wilk. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52(3-4):591–611, 1965.

[SW05] Penelope Sweetser and Peta Wyeth. GameFlow: A Model for Evaluating Player Enjoyment in Games. *ACM Computers in Entertainment*, 3(3), 2005.

[SWMW09] Sergey Smirnov, Matthias Weidlich, Jan Mendling, and Mathias Weske. Action Patterns in Business Process Models. In *Proceedings of the 7th International Joint Conference on Service-Oriented Computing (ICSOC-ServiceWave 2009)*, pages 115–129, Stockholm, Sweden, 2009.

[SZ03] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals.* The MIT Press, Cambridge, MA, United States, 1st edition, 2003.

[Szu05] Dave Szulborski. *This Is Not A Game: A Guide to Alternate Reality Gaming.* Lulu.com, Raleigh, NC, United States, 1st edition, 2005.

[Tam87] Roberto Tamassia. Embedding a Graph in the Grid With the Minimum Number of Bends. *SIAM Journal of Computing*, 16(3):421–444, 1987.

[Tan87] J. S. Tanaka. "How Big Is Big Enough?" Sample Size and Goodness of Fit in Structural Equation Models with Latent Variables Author. *Child Development*, 58(1):134–146, 1987.

[TD11] Mohsen Tavakol and Reg Dennick. Making Sense of Cronbach's Alpha. *International Journal of Medical Education*, 2:53–55, 2011.

[Tha00] Bernhard Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2000.

[TLB14] Scott Thiebes, Sebastian Lins, and Dirk Basten. Gamifying Information Systems – A Synthesis of Gamification Mechanics and Dynamics. In *Proceedings of the 22nd European Conference on Information Systems (ECIS 2014)*, Tel Aviv, Israel, 2014.

[TM07b] Manfred Thüring and Sascha Mahlke. Usability, Aesthetics and Emotions in Human–Technology Interaction. *International Journal of Psychology*, 42(4):253–264, 2007.

[Trk10] Peter Trkman. The Critical Success Factors of Business Process Management. *International Journal of Information Management*, 30(2):125–134, 2010.

[vA06] Luis von Ahn. Games with a Purpose. *Computer*, 39(6):92–94, 2006.

[vA09] Luis von Ahn. Human Computation. In *Proceedings of the 46th Annual Design Automation Conference (DAC 2009)*, pages 418–419, San Francisco, CA, United States, 2009.

[vA13] Luis von Ahn. Duolingo: Learn a Language for Free while Helping to Translate the Web. In *Proceedings of the 18th International Conference on Intelligent User Interfaces (IUI 2013)*, pages 1–2, Santa Monica, CA, United States, 2013.

[vAD08] Luis von Ahn and Laura Dabbish. Designing Games with a Purpose. *Communications of the ACM*, 51(8):58–67, 2008.

[Van06] Richard Van Eck. Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless. *EDUCAUSE Review*, 41(2):16–30, 2006.

[VB08] Viswanath Venkatesh and Hillol Bala. Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences*, 39(2):273–315, 2008.

[vBBB+11] Jan vom Brocke, Jörg Becker, Alessio Maria Braccini, Rimantas Butleris, Birgit Hofreiter, Kestutis Kapocius, Marco De Marco, Günter Schmidt, Stefan Seidel, Alexander Simons, Tomas Skopal, Armin Stein, Stefan Stieglitz, Reima Suomi, Gottfried Vossen, Robert Winter, and Stanislaw Wrycza. Current and Future Issues in BPM Research: A European Perspective from the ERCIS Meeting 2010. *Communications of the Association for Information Systems*, 28:393–414, 2011.

[vBHPV07] Patrick van Bommel, Stijn J. B. A. Hoppenbrouwers, Henderik A. Proper, and Theo P. Van Der Weide. QoMo: A Modelling Process Quality Framework Based on SEQUAL. In *Proceedings of the 12th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2007)*, pages 118–127, Trondheim, Norway, 2007.

[vBS11] Jan vom Brocke and Theresa Sinnl. Culture in Business Process Management: A Literature Review. *Business Process Management Journal*, 17(2):357–377, 2011.

[vBSN+09] Jan vom Brocke, Alexander Simons, Björn Niehaves, Kai Riemer, Ralf Plattfaut, and Anne Cleven. Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature. In *Proceedings of the 17th European Conference on Information Systems (ECIS 2009)*, pages 2206–2217, Verona, Italy, 2009.

[vBSR+14] Jan vom Brocke, Theresa Schmiedel, Jan C. Recker, Peter Trkman, Willem Mertens, and Stijn Viaene. Ten Principles of Good Business Process Management. *Business Process Management Journal*, 20(4):530–548, 2014.

[VCM+07] Irene Vanderfeesten, Jorge Cardoso, Jan Mendling, Hajo A. Reijers, and Wil M. P. van der Aalst. Quality Metrics for Business Process Models. In Layna Fischer, editor, *2007 BPM and Workflow Handbook*, chapter 17, pages 179–190. Future Strategies Inc., Lighthouse Point, FL, United States, 2007.

[VD13] Pieter Van Gorp and Remco M. Dijkman. A Visual Token-Based Formalization of BPMN 2.0 Based on In-Place Transformations. *Information and Software Technology*, 55(2):365–394, 2013.

[vdA98] Wil M. P. van der Aalst. The Application of Petri Nets To Workflow Management. *Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[vdA99] Wil M. P. van der Aalst. Formalization and Verification of Event-Driven Process Chains. *Information and Software Technology*, 41(10):639–650, 1999.

[vdA11] Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancment of Business Processes.* Springer, Berlin, Heidelberg, Germany, 1st edition, 2011.

[vdA12a] Wil M. P. van der Aalst. A Decade of Business Process Management Conferences: Personal Reflections on a Developing Discipline. In *Proceedings of the 10th International Conference on Business Process Management (BPM 2012)*, pages 1–16, Tallinn, Estonia, 2012.

[vdA12b] Wil M. P. van der Aalst. What Makes a Good Process Model? Lessons Learned from Process Mining. *Software and Systems Modeling*, 11(4):557–569, 2012.

[vdA13] Wil M. P. van der Aalst. Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, 2013:1–37, 2013.

[vdABtHK00] Wil M. P. van der Aalst, Alistair P. Barros, Arthur H. M. ter Hofstede, and Bartek Kiepuszewski. Advanced Workflow Patterns. In *Proceedings of the 7th International Conference on Cooperative Information Systems (CoopIS 2000)*, pages 18–29, Eilat, Israel, 2000.

[vdALR15] Han van der Aa, Henrik Leopold, and Hajo A. Reijers. Detecting Inconsistencies Between Process Models and Textual Descriptions Han. In *Proceedings of the 13th International Conference on Business Process Management (BPM 2015)*, pages 90–105, Innsbruck, Austria, 2015.

[vdARS05] Wil M. P. van der Aalst, Hajo A. Reijers, and Minseok Song. Discovering Social Networks from Event Logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593, 2005.

[vdAS11] Wil M. P. van der Aalst and Christian Stahl. *Modeling Business Processes: A Petri Net-Oriented Approach.* The MIT Press, Cambridge, MA, United States, 1st edition, 2011.

[vdAtHKB03] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, and Alistair P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.

[vdAvH02] Wil M. P. van der Aalst and Kees M. van Hee. *Workflow Management: Models, Methods, and Systems.* The MIT Press, Cambridge, MA, United States, 1st edition, 2002.

[vdAvHtH+11] Wil M. P. van der Aalst, Kees M. van Hee, Arthur H. M. ter Hofstede, N. Sidorova, H. M. W. Verbeek, M. Voorhoeve, and M. T. Wynn. Soundness of Workflow Nets: Classification, Decidability, and Analysis. *Formal Aspects of Computing*, 23(3):333–363, 2011.

[vdAWG05] Wil M. P. van der Aalst, Mathias Weske, and Dolf Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data & Knowledge Engineering*, 53(2):129–162, 2005.

[vdKK15] Bart van der Kruys and Vassilis Javed Khan. Kindle: How Gamification Can Motivate Jobseekers. In *Proceedings of the 2nd International Conference on HCI in Business (HCIB 2015)*, pages 481–492, Los Angeles, CA, United States, 2015.

[Ven00] Viswanath Venkatesh. Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion Into the Technology Acceptance Model. *Information Systems Research*, 11(4):342–365, 2000.

[VF12] Simon Vogt and Andreas Fink. Using Status Feeds for Peer Production by Coordinating Non-predictable Business Processes. In *Proceedings of the 4th Workshop on Business Process Management and Social Software (BPMS2 2011)*, pages 253–265, Clermont-Ferrand, France, 2012.

[VFL10] Dominik Vanderhaeghen, Peter Fettke, and Peter Loos. Organisations- und Technologieoptionen des Geschäftsprozessmanagements aus der Perspektive des Web 2.0. *Wirtschaftsinformatik*, 52(1):17–32, 2010.

[VH07] Gottfried Vossen and Stephan Hagemann. *Unleashing Web 2.0: From Concepts to Creativity.* Morgan Kaufmann, Burlington, MA, United States, 1st edition, 2007.

[vLKS+11] Tatiana von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[VMDD03] Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3):425–478, 2003.

[vRKH15] Mark von Rosing, Neil Kemp, and Maria Hove. Process Tagging - A Process Classification and Categorization Concept. In Mark van Rosing, August-Wilhelm Scheer, and Henrik van Scheel, editors, *The Complete Business Process Handbook*, chapter 8, pages 123–171. Morgan Kaufmann Publishers, Waltham, MA, United States, 1st edition, 2015.

[VRM+08] Irene Vanderfeesten, Hajo A. Reijers, Jan Mendling, Wil M. P. van der Aalst, and Jorge Cardoso. On a Quest for Good Process Models: The Cross-Connectivity Metric. In *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008)*, pages 480–494, Montpellier, France, 2008.

[VW86] Iris Vessey and Ron Weber. Structured Tools and Conditional Logic: An Empirical Investigation. *Communications of the ACM*, 29(1):48–57, 1986.

[Wad01] Vance Waddle. Graph Layout for Displaying Data Structures. In *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, pages 241–252, Colonial Williamsburg, VA, United States, 2001.

[War04] Collin Ware. *Information Vizualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA, United States, 2nd edition, 2004.

[WD14] Steffen P. Walz and Sebastian Deterding. *The Gameful World: Approaches, Issues, Applications*. The MIT Press, Cambridge, MA, United States, 1st edition, 2014.

[Wei11] Burkhard Weiß. *Process Modelling and Analysis in Banks: Leveraging Business Process Optimization in the Financial Sector*. Doctoral thesis, University of Münster, 2011.

[Wer14] Kevin Werbach. (Re)Defining Gamification: A Process Approach. In *Proceedings of the 9th International Conference on Persuasive Technology (PERSUASIVE 2014)*, pages 266–272, Padua, Italy, 2014.

[Wes07] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin, Heidelberg, Germany, 1st edition, 2007.

[WGK10] Matthew O. Ward, Georges G. Grinstein, and Daniel A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. Taylor & Francis, London, United Kingdom, 1st edition, 2010.

[WH12] Kevin Werbach and Dan Hunter. *For the Win: How Game Thinking Can Revolutionize Your Bussiness*. Wharton Digital Press, Philadelphia, PA, United States, 1st edition, 2012.

[Whi59] Robert W. White. Motivation Reconsidered: The Concept of Competence. *Psychological Review*, 66(5):297–333, 1959.

[Wil45] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[WKH15] Yue Wu, Atreyi Kankanhalli, and Ke-Wei Huang. Gamification in Fitness Apps: How Do Leaderboards Influence Exercise? In *Proceedings of the 36th International Conference on Information Systems (ICIS 2015)*, Fort Worth, TX, United States, 2015.

[WL13] Jiming Wu and Xinjian Lu. Effects of Extrinsic and Intrinsic Motivators on Using Utilitarian, Hedonic, and Dual-Purposed Information Systems: A Meta-Analysis. *Journal of the Association for Information Systems*, 14(3):153–191, 2013.

[WL16] Ronald L. Wasserstein and Nicole A. Lazar. The ASA's Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, 70(2):129–133, 2016.

[WRMR11] Barbara Weber, Manfred Reichert, Jan Mendling, and Hajo A. Reijers. Refactoring Large Process Model repositories. *Computers in Industry*, 62(5):467–486, 2011.

[WvNvOvdS13] Pieter Wouters, Christof van Nimwegen, Herre van Oostendorp, and Erik D. van der Spek. A Meta-Analysis of the Cognitive and Motivational Effects of Serious Games. *Journal of Educational Psychology*, 105(2):249–265, 2013.

[WW90] Yair Wand and Ron Weber. An Ontological Model of an Information System. *IEEE Transactions on Software Engineering*, 16(11):1282–1292, 1990.

[WW02a] Yair Wand and Ron Weber. Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda. *Information Systems Research*, 13(4):363–376, 2002.

[WW02b] Jane Webster and Richard T. Watson. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2):xiii–xxiii, 2002.

[YPW14] Alfa R. Yohannis, Yulius D. Prabowo, and Alexander Waworuntu. Defining Gamification: From Lexical Meaning and Process Viewpoint Towards a Gameful Reality. In *Proceedings of the 2014 International Conference on Information Technology Systems and Innovation (ICITSI 2014)*, pages 284–289, Bandung, Bali, 2014.

[ZC11] Gabe Zichermann and Christopher Cunningham. *Gamification By Design*. O'Reilly, Sebastopol, Canada, 1st edition, 2011.

[ZCZ10] Jose Zagal, Susy S. Chan, and Jingli Zhang. Measuring Flow Experience of Computer Game Players. In *Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010)*, Lima, Peru, 2010.

[ZFC11] Jingli Zhang, Xiaowen Feng, and Susy S. Chan. Development of an Instrument to Measure Flow Experience in Computer Game Play. In *Proceedings of the 9th Conference of the AIS Special Interest Group on Human-Computer Interaction (SIGHCI 2011)*, Shanghai, China, 2011.

[Zha08] Ping Zhang. Motivational Affordances: Reasons for ICT Design and Use. *Communications of the ACM*, 51(11):145–147, 2008.

[Zim04] Donald W. Zimmerman. Inflation of Type I Error Rates by Unequal Variances Associated with Parametric, Nonparametric, and Rank-Transformation Tests. *Psicologica*, 25(1):103–133, 2004.

[ZL13] Gabe Zichermann and Joselin Linder. *The Gamification Revolution: How Leaders Leverage Game Mechanics to Crush the Competition.* McGraw-Hill Education, New York City, NY, United States, 1st edition, 2013.

[zM04] Michael zur Muehlen. *Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-based Process Information Systems.* Logos, Berlin, Germany, 2004.

[zMR08] Michael zur Muehlen and Jan C. Recker. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008)*, pages 465–479, Montpellier, France, 2008.

List of Web Pages

[1] Anderson, Janna ; Rainie, Lee: *The Future of Gamification*. http://www.pewinternet.org/2012/05/18/the-future-of-gamification/. Version: May 2012. – Last accessed: 2016-11-25.

[2] Bitkom: *Gaming hat sich in allen Altersgruppen etabliert*. https://www.bitkom.org/Presse/Presseinformation/Gaming-hat-sich-in-allen-Altersgruppen-etabliert.html. Version: July 2015. – Last accessed: 2016-09-14.

[3] Bogost, Ian: *Gamification is Bullshit: My Position Statement at the Wharton Gamification Symposium*. http://bogost.com/writing/blog/gamification_is_bullshit/. Version: August 2011. – Last accessed: 2016-11-25.

[4] Bogost, Ian: *Persuasive Games: Exploitationware*. http://www.gamasutra.com/view/feature/134735/persuasive_games_exploitationware.php. Version: May 2011. – Last accessed: 2016-11-25.

[5] Burke, Brian: *Gartner Redefines Gamification*. http://blogs.gartner.com/brian_burke/2014/04/04/gartner-redefines-gamification/. Version: April 2014. – Last accessed: 2016-07-02.

[6] Chaplin, Heather: *I Don't Want To Be a Superhero*. http://www.slate.com/articles/technology/gaming/2011/03/i_dont_want_to_be_a_superhero.html. Version: March 2011. – Last accessed: 2016-11-25.

[7] Deterding, Sebastian: *A Quick Buck by Copy and Paste*. http://gamification-research.org/2011/09/a-quick-buck-by-copy-and-paste/. Version: September 2011. – Last accessed: 2016-11-25.

[8] Deterding, Sebastian: *Gamification by Design: Response to O'Reilly*. http://gamification-research.org/2011/09/gamification-by-design-response-to-oreilly/. Version: September 2011. – Last accessed: 2016-11-25.

[9] Duolingo: *Duolingo: Learn Languages Free*. https://play.google.com/store/apps/details?id=com.duolingo. Version: August 2016. – Last accessed: 2016-08-30.

[10] French, Michael: *Inside Rockstar North - Part 2: The Studio*. http://www.develop-online.net/studio-profile/inside-rockstar-north-part-2-the-studio/0184061. Version: October 2013. – Last accessed: 2016-11-25.

[11] GARTNER: *Gartner Says By 2015, More Than 50 Percent of Organizations That Manage Innovation Processes Will Gamify Those Processes*. http://www.gartner.com/newsroom/id/1629214. Version: April 2011. – Last accessed: 2016-11-23.

[12] GARTNER: *Gartner's 2011 Hype Cycle Special Report Evaluates the Maturity of 1,900 Technologies*. http://www.gartner.com/newsroom/id/1763814. Version: August 2011. – Last accessed: 2016-11-23.

[13] GARTNER: *Gartner Says by 2014, 80 Percent of Current Gamified Applications Will Fail to Meet Business Objectives Primarily Due to Poor Design*. http://www.gartner.com/newsroom/id/2251015. Version: November 2012. – Last accessed: 2016-11-23.

[14] GARTNER: *Gartner Says Organizational Politics Will Prevent at Least One-Third of BPM Efforts Through 2016*. http://www.gartner.com/newsroom/id/1914714. Version: February 2012. – Last accessed: 2016-11-23.

[15] GARTNER: *Gartner's 2012 Hype Cycle for Emerging Technologies Identifies "Tipping Point" Technologies That Will Unlock Long-Awaited Technology Scenarios*. http://www.gartner.com/newsroom/id/2124315. Version: August 2012. – Last accessed: 2016-11-23.

[16] GARTNER: *Gartner's 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relationship Between Humans and Machines*. http://www.gartner.com/newsroom/id/2575515. Version: August 2013. – Last accessed: 2016-11-23.

[17] KREIMEIER, Bernd: *The Case For Game Design Patterns*. http://www.gamasutra.com/view/feature/132649/the_case_for_game_design_patterns.php. Version: March 2002. – Last accessed: 2016-11-17.

[18] LAZZARO, Nicole: *Why We Play Games: Four Keys to More Emotion Without Story*. http://dx.doi.org/10.1111/j.1464-410X.2004.04896.x. Version: March 2004. – Last accessed: 2017-04-04.

[19] LISCHKA, Konrad: *Kriegsspiel: Wie preußische Militärs den Rollenspiel-Ahnen erfanden*. http://www.spiegel.de/netzwelt/spielzeug/kriegsspiel-wie-preussische-militaers-den-rollenspiel-ahnen-erfanden-a-625745.html. Version: June 2009. – Last accessed: 2017-02-03.

[20] MARCZEWSKI, Andrzej: *Understanding Intrinsic Motivation with RAMP*. http://www.gamification.co/2013/05/01/understanding-intrinsic-motivation-with-ramp/. Version: May 2013. – Last accessed: 2016-11-28.

[21] MCGONIGAL, Jane: *Be a Gamer, Save the World*. http://www.wsj.com/articles/SB10001424052748704590704576092460302990884. Version: January 2011. – Last accessed: 2016-09-14.

[22] MCLAUGHLIN, Martyn: *New GTA V Release Tipped to Rake in £1bn in Sales*. http://www.scotsman.com/lifestyle/gadgets-gaming/new-gta-v-release-tipped-

to-rake-in-1bn-in-sales-1-3081943. Version: September 2013. – Last accessed: 2016-11-25.

[23] Merriam-Webster: *Definition of Gamification*. http://www.merriam-webster.com/dictionary/gamification. Version: April 2017. – Last accessed: 2017-04-04.

[24] Nike: *Nike+ Fuel on the App Store*. https://itunes.apple.com/us/app/id493325070. Version: September 2015. – Last accessed: 2016-08-03.

[25] Nike: *Nike+ Move on the App Store*. https://itunes.apple.com/us/app/nike+-move/id712498492. Version: February 2015. – Last accessed: 2016-08-03.

[26] Oxford English Dictionary: *process, n.* http://www.oed.com/view/Entry/151794. Version: June 2007. – Last accessed: 2017-02-24.

[27] Oxford English Dictionary: *quality, n. and adj.* http://www.oed.com/view/Entry/155878. Version: December 2007. – Last accessed: 2016-10-11.

[28] Oxford English Dictionary: *framework, n.* http://www.oed.com/view/Entry/74161. Version: September 2011. – Last accessed: 2017-01-09.

[29] Protalinski, Emil: *100M users Strong, Duolingo Raises 45M USD Led by Google at a 470M USD Valuation to Grow Language-Learning Platform*. http://venturebeat.com/2015/06/10/100m-users-strong-duolingo-raises-45m-led-by-google-at-a-470m-valuation-to-grow-language-learning-platform/. Version: June 2015. – Last accessed: 2017-03-06.

[30] Radoff, Jon: *Gamification, Behaviorism and Bullshit*. http://web.archive.org/web/20131111025214/http://radoff.com/blog/2011/08/09/gamification-behaviorism-bullshit/. Version: August 2011. – Last accessed: 2016-11-25.

[31] Rey, PJ: *Gamification, Playbor and Exploitation*. https://thesocietypages.org/cyborgology/2012/10/15/gamification-playbor-exploitation-2/. Version: October 2012. – Last accessed: 2016-11-25.

[32] Robertson, Margaret: *Can't Play, Won't Play*. http://hideandseek.net/2010/10/06/cant-play-wont-play/. Version: October 2010. – Last accessed: 2016-11-25.

[33] Sheffield, Brandon: *Opinion: Let's Retire the Word 'Gamer'*. http://www.gamasutra.com/view/news/192107/Opinion_Lets_retire_the_word_gamer.php. Version: May 2013. – Last accessed: 2017-02-03.

[34] Stuart, Keith: *Seven Reasons Why Grown Ups Should Play More Video Games*. https://www.theguardian.com/technology/2017/jan/05/why-grown-ups-should-play-more-video-games. Version: January 2017. – Last accessed: 2017-02-03.

[35] Swarm: *Swarm is the App that Turns Every Day Into a Game*. https://www.swarmapp.com. Version: 2017. – Last accessed: 2017-03-27.

[36] The Nielsen Company: *Multi-Platform Gaming: For the Win!* http://www.nielsen.com/us/en/insights/news/2014/multi-platform-gaming-for-the-win.html. Version: May 2014. – Last accessed: 2016-09-14.

[37] The NPD Group: *The NPD Group Reports 34 Million Core Gamers Spend an Average of 22 Hours per Week Playing Video Games.* https://www.npd.com/wps/portal/npd/us/news/press-releases/the-npd-group-reports-34-million-core-gamers-spend-an-average-of-22-hours-per-week-playing-video-games/. Version: May 2014. – Last accessed: 2016-09-14.

[38] Vogel, Lars: *Eclipse RCP based on the 3.x API - Tutorial.* http://www.vogella.com/tutorials/Eclipse3RCP/article.html. Version: July 2016. – Last accessed: 2017-04-04.

[39] Vogel, Lars: *Eclipse RCP (Rich Client Platform) - Tutorial.* http://www.vogella.com/tutorials/EclipseRCP/article.html. Version: July 2016. – Last accessed: 2017-04-04.

[40] Wawro, Alex: *Gabe Newell's Game Design Advice: 'Get Into an Iteration Cycle'.* http://www.gamasutra.com/view/news/289434/Gabe_Newells_game_design_advice_Get_into_an_iteration_cycle.php. Version: January 2017. – Last accessed: 2017-03-11.

List of Abbreviations

3QM Quality Marks, Metrics, and Measurement.

7PMG Seven Process Modeling Guidelines.

AOM Asset Oriented Modeling.

ARIS Architecture of Integrated Information Systems.

AVE Average Variance Extracted.

BPEL Business Process Execution Language.

BPM Business Process Management.

BPMN Business Process Model and Notation.

BPMS Business Process Management System.

CEGE Core Elements of the Gaming Experience.

CFA Confirmatory Factor Analysis.

CMQF Conceptual Model Quality Framework.

CSCW Computer-Supported Cooperative Work.

CUE Components of User Experience.

DBIS Databases and Information Systems.

DFS-2 Dispositional Flow Scale-2.

DSRM Design Science Research Methodology.

EBNF extended Backus-Naur form.

EFA Exploratory Factor Analysis.

EPC Event-driven Process Chain.

ERCIS European Research Center for Information Systems.

ERP Enterprise Resource Planning.

ESA Entertainment Software Association.

FIFO First In First Out.

FKS Flow-Kurzskala.

FPS First Person Shooter.

FSS-2 Flow State Scale-2.

GoM Guidelines of Modeling.

GWAP Game with a Purpose.

HBM Horus Business Modeler.

HERM Higher Order Entity Relationship Model.

iBPMS intelligent Business Process Management System.

IDE Integrated Development Environment.

IMI intrinsic motivation inventory.

IS Information Systems.

IT Information Technology.

KPI Key Performance Indicator.

MDA Mechanics, Dynamics, Aesthetics.

MDI Multiple Document Interface.

MMORPG Massively Multiplayer Online Role-playing Game.

MUD Multi-User Dungeon.

PBL Points, Badges, Leaderboards.

PENS Player Experience of Need Satisfaction.

QoMo Quality of Modelling.

RCP Rich Client Platform.

RDBMS Relational Database Management System.

RPG Role-playing Game.

SDT Self-determination Theory.

SEQUAL Semiotic Quality Model.

SHM Semantic Hierarchy Model.

SIQ Simple, Integrates, Quality.

Social BPM Social Business Process Management.

SWOT Strengths, Weaknesses, Opportunities, Threats.

TAM Technology Acceptance Model.

TAM3 Technology Acceptance Model 3.

TRA Theory of Reasoned Action.

UML Unified Modeling Language.

US United States.

UTAUT Unified Theory of Acceptance and Use of Technology.

WfMS Workflow Management System.

WSDL Web Services Description Language.

WWU Westfälische Wilhelms-Universität.

XML Extensible Markup Language.

Ludography

— Gamification

Ariadne PathLogin (2016). Authentication system published by Ebbers and Brune. Source: [EB16]. *Ariadne PathLogin is a gamified authentication method in which users need to complete a simple game to authenticate themselves instead of providing a password. The system requires users to choose a unique user name and a playing figure, and to define a path across a playing board that serves as the equivalent to the password.* 120, 122

bant (2012). Diabetes self-management app published by Cafazzo et al.. Source: [CCH+12]. *bant is a gamified diabetes self-management smartphone application for adolescents whose main purpose it is to increase the frequency of conducted blood glucose readings provided via a Bluetooth dongle. Rewards for doing so include virtual points and vouchers for smartphone apps.* 120, 122

Beehive (2008). Corporate social network published by IBM. Website: http://researcher.watson.ibm.com/researcher/view\protect_group.php?id=1231. Last accessed: 2017-04-21. *Beehive was a corporate social networking site operated by IBM for its own employees which allowed users to maintain a profile page, share photos and lists, connect with others, and write comments. It was extended with gamification functionality by Farzan et al., who integrated a points system and a leaderboard into the site [FDM+08].* 133

emoticon-bin (2013). Recyling bin published by Berengues et al.. Source: [BAZN13]. *emoticon-bin is a gamified recycling bin for plastic bottles that rewards users with a happy smiley face and a coin sound whenever a sensor detects that a bottle is inserted.* 119, 120, 122

Foursquare (2009). Search-and-discovery service published by Foursquare Labs. Website: http://foursquare.com/. Last accessed: 2017-04-20. *Foursquare is a local search and recommender service for facilities such as restaurants, cafés, and night clubs. Traditionally, it was a social network that provided specific features based on the location data of its users and whose mobile application was built on gamification functionality.* 117, 118

Kindle (2015). Job-seeking platform published by van der Kruys and Khan. Source: [vdKK15]. *Kindle is a gamification concept intended to motivate job seekers to look for work. To that extent, an online system is provided that allows users to upload their curriculum vitae as the central game object, the main goal of the platform being quality improvement.* 120, 122

Quick Quiz (2013). Multiple-choice quiz platform published by Cheong et al.. Source: [CCF13]. *Quick Quiz is a gamified multiple-choice quiz software that allows players to earn points for answering questions correctly under time pressure, with the height of the reward varying with answer speed. Furthermore, a leaderboard enables competition between different players.* 119, 122

WeBWorK (1994). Homework submission system published by Mathematical Association of America. Website: http://webwork.maa.org/. Last accessed: 2017-04-21. *WeBWorK is an online homework submission system with a focus on mathematics courses. It was extended with gamification functionality by Goehle, who integrated an experience points system and achievements into the site [Goe13].* 121, 122

Duolingo (2011). Language-learning app published by Duolingo. Website: https://www.duolingo.com/. Last accessed: 2017-04-20. *Duolingo is a language-learning Web application with accompanying smartphone apps that employs gamification as a central strategy for player motivation. With Duolingo, users can learn a wide variety of different languages such as Spanish, French, English, or German by applying their reading, listening, and speaking skills.* 118, 119, 136, 212, 234, 359

Nike+ (2006). Fitness community published by Nike. Website: http://nikeplus.nike.com/nikeplus/. Last accessed: 2017-04-20. *Nike+ is a gamified fitness community based on various physical products and smartphone apps. Its basic idea is the use of an activity tracker, the so-called Nike+ FuelBand, that measures the everyday physical activity of its wearer and converts it into a points score called NikeFuel.* 116, 117, 137, 158, 211

Stack Exchange (2009). Gamified Q&A Platform published by Stack Exchange. Website: https://stackexchange.com/. Last accessed: 2017-04-21. *Stack Exchange is a gamified question-and-answer platform in which users can create questions on various topics, most notably in technology-related areas such as programming and network administration. It employs a points-based reputation system through which users can earn points depending on the quality of their questions and responses.* 232

YouTube Heroes (2016). Gamified content moderation published by YouTube. Website: https://support.google.com/youtube/answer/7124236. Last accessed: 2017-04-21. *YouTube Heroes is a gamified approach for content moderation that allows users to earn points for reporting videos that violate the guidelines of YouTube. By earning enough points, users can increase their level, thereby unlocking certain privileges and premium features.* 232

— Platforms

PlayStation Network (2006). Online service published by Sony Interactive Entertainment. Website: http://www.playstationnetwork.com/. Last accessed: 2016-12-12. *The PlayStation Network is Sony's media entertainment platform for its consoles and other devices. Besides features for online gaming, it also comprises a marketplace, movie straming, music streaming, and cloud gaming facilities.* 130

Steam (2003). Game distribution platform published by Valve Corporation. Website: http://store.steampowered.com/. Last accessed: 2016-12-12. *Steam is a distribution platform for digital games that offers facilities for online gaming, social networking, digital rights management, and video streaming.* 130

Xbox Live (2002). Online service published by MICROSOFT. Website: http://xbox.com/live/. Last accessed: 2016-12-12. *Xbox Live is Microsoft's online gaming and media entertainment platform for its consoles and further devices. Besides features for online gaming, it also comprises a marketplace, various apps for video and music streaming, and features for social networking.* 130, 237

— SERIOUS GAMES

ESP Game (2006). Image tagging game published by VON AHN. Source: [vA06]. *The ESP Game is a human computation game presenting the task of tagging images as a playful activity. Players are randomly paired with others, and must guess which labels the other person will use to describe the current image.* 123

Foldit (2008). Puzzle game published by UNIVERSITY OF WASHINGTON. Source: [CKT+10]. *Foldit is an online multiplayer game in which players collaborate and compete in optimizing protein structures, thereby becoming amateur scientists.* 111, 123, 124

Kriegsspiel (1812). Military training game published by GEORG VON REISSWITZ. Source: [19]. *The Kriegsspiel served the purpose of training Prussian army personnel in tactical battle maneuvers by letting them play a game that can be considered an ancestor of modern tabletop role-playing games.* 3

Peekaboom (2006). Image tagging game published by VON AHN. Source: [vA06]. *Peekaboom is related to the ESP Game, but aims to enrich the collected data by incorporating the locations at which the tagged objects appears in an image into gameplay.* 124

— VIDEOGAMES

Age of Empires II (1999). Real-time strategy game published by ENSEMBLE STUDIOS. Website: https://www.ageofempires.com/games/aoeii/. Last accessed: 2017-04-21. *Age of Empires II is a strategy game set in the Middle Ages in which players can choose from 13 civilizations and 5 campaigns*

based on historic events. The main goal for players is to collect resources to develop their civilization, and to ultimately defeat their enemies on the battlefield. 220

Castlevania: Symphony of the Night (1997). Action-adventure role-playing game published by Konami. Website: https://www.konami.com/. Last accessed: 2016-12-06. *In this game, players assume the role of Alucard and are tasked with exploring Dracula's castle and defeating its lord. To that extent, they must battle a variety of monsters, progressively unlock new areas of the castle, and collect weapons and other equipment items along the way.* 135, 238

Diablo III (2012). Action role-playing game published by Blizzard Entertainment. Website: http://us.battle.net/d3/en/. Last accessed: 2016-12-02. *In the game, players choose one of six character classes – Barbarian, Crusader, Demon Hunter, Monk, Witch Doctor or Wizard (with the Crusader being unavailable unless the player has purchased the expansion pack, Diablo III: Reaper of Souls) – and are tasked with defeating the Lord of Terror, Diablo.* 141, 218, 226, 238

Final Fantasy IX (2000). Role-playing game published by Square. Website: http://www.jp.square-enix.com/ff9/en/. Last accessed: 2017-04-21. *In this game, players take part in the war between two nations, and are ultimately tasked with defeating the queen of one of the two. To that extent, they must build a party of warriors, navigate an extensive game world, communicate with non-player characters, and successfully complete dungeons populated with monsters.* 128, 129, 226, 238

Grand Theft Auto V (2013). Open-world action-adventure game published by Rockstar Games. Website: www.rockstargames.com/V/. Last accessed: 2017-04-21. *In this game, players follow the lives of three criminals and can freely explore an extensive open world or complete missions to progress through its story. Important mechanics of the game include a battle system, acquiring and driving vehicles, and avoiding capture by law enforcement.* 195

Pac-Man (1980). Arcade game published by Namco. Website: http://pacman.com/. Last accessed: 2017-04-21. *In this game, players must navigate the eponymous Pac-Man through a maze, collecting points by eating small dots while avoiding hostile ghosts. When touching a ghost, players lose a life, and if all lives are lost, the game ends. Inversely, collecting larger dots called power pellets awards players with the temporary ability to eat enemies.* 128, 129

Pokémon Go (2016). Augmented reality game published by Niantic. Website: http://www.pokemongo.com/. Last accessed: 2017-04-21. *In this location-based game, players register their movements via the GPS sensors of their smartphones, and are presented with Pokémon to capture depending on their location. The game mirrors a subset of the features found in the conventional Pokémon games, but aims to transfer them to the "real world".* 125

Pong (1972). Arcade game published by Atari. Website: https://www.atari.com/. Last accessed: 2017-04-21. *Pong is a table tennis game in which two players control paddles at separate ends of the playing field. The players must hit a ball back and forth until one of them fails to do so, and the other gains a point.* 1

The Elder Scrolls V: Skyrim (2011). Open-world role-playing game published by Bethesda. Website: https://elderscrolls.bethesda.net/skyrim/. Last accessed: 2017-04-21. *In this game, players may freely explore the land of Skyrim and the towns, caves, fortresses, and dungeons it contains. While following the overarching storyline, they must defeat various human enemies and monsters, collect weapons, learn new skills and magic spells, and complete quests given by non-player characters.* 234

Super Mario World (1990). Jump and Run published by Nintendo. Website: https://www.nintendo.co.jp/n02/shvc/mw/index.html. Last accessed: 2017-04-21. *In this game, players must restore peace to Dinosaur Land and rescue Princess Peach from Bowser by navigating the player character, Mario, through up to 74 levels from left to right. The basic mechanics of the game consist of walking, running, and jumping, which the player* *must combine to reach the end of each level while overcoming various obstacles.* xiv, 135–137, 159, 239

Spacewar! (1962). Space combat game published by Steve Russell. Website: http://www.computerhistory.org/pdp-1/spacewar/. Last accessed: 2017-04-21. *In this game, two players take control of separate spaceships and try to destroy the vessel of their enemy while navigating around a star. The game features limitations of fuel and torpedos, as well as a realistic physics system.* 1

Tetris (1984). Puzzle game published by Alexey Pajitnov. Website: http://www.tetris.com/. Last accessed: 2017-04-21. *In Tetris, players must steer and rotate shapes of various sizes falling down onto the playing field so that the screen fills up as slowly as possible. To prolong the game, players must fill up horizontal lines, upon which they are removed. The more lines the player manages to delete with a single shape, the more points he is awarded. Once there is no further vertical space for additional shapes, the game terminates.* 133

The Witcher 3: Wild Hunt (2015). Action role-playing game published by CD Projekt RED. Website: http://thewitcher.com/en/witcher3/. Last accessed: 2017-04-21. *In this open-world game, players control Geralt of Rivia, a monster hunter known as a witcher. Similarly to other games of this type, players need to explore the vast world, defeat enemies, collect equipment, learn new skills, and solve quests to follow the overall storyline.* 234

VVVVVV (2010). 2D Platformer published by Terry Cavanagh. Website: http://thelettervsixtim.es. Last accessed: 2017-04-21. *VVVVVV is a 2D platforming game in which players must explore a foreign planet while avoiding hazards such as spikes and hostile objects. Besides movement, the only other action that players can perform and on which most challenges and puzzles are based is to reverse the direction of gravity.* 195

World of Warcraft (2004). Massively multiplayer online role-playing game published by Blizzard Entertainment. Website: https://worldofwarcraft.

com/en-us/. Last accessed: 2017-04-21. *In this game, players control an individual character and are tasked with exploring the game world, defeating various enemies, and completing quests given by non-player characters. Being an online game, a considerable portion of the gameplay is based on the cooperation with other players, with whom they can form guilds.* 2, 219

Gamification for Business Process Modeling

Nicolas Pflanzl

Gamification is a novel, industry-driven trend that is concerned with the application of game design elements to non-game contexts. The main goal of this work is to examine how gamification can be used to support business process modeling. To that extent, design science research was conducted to determine current challenges and problems in business process management that may be solved through gamification. To address these issues, a "gamified" version of the Horus Business Modeler – a software for process modeling – was designed, implemented, and evaluated. The results of the evaluation are mixed and show increases in model quality and tool assessment, but no impact on motivation. Overall, this work contributes to research by providing insights into the impacts of gamification in a domain not previously discussed, and by showing how certain issues of business process management can be addressed by means of an IT artifact with particular characteristics.

50,00 €
ISBN 978-3-8405-0176-0